

Lecture Notes on Data Engineering
and Communications Technologies 149



Sergii Babichev
Volodymyr Lytvynenko *Editors*

Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making

2022 International Scientific
Conference “Intellectual Systems
of Decision-Making and Problems
of Computational Intelligence”,
Proceedings

Lecture Notes on Data Engineering and Communications Technologies

Volume 149

Series Editor

Fatos Xhafa, Technical University of Catalonia, Barcelona, Spain

The aim of the book series is to present cutting edge engineering approaches to data technologies and communications. It will publish latest advances on the engineering task of building and deploying distributed, scalable and reliable data infrastructures and communication systems.

The series will have a prominent applied focus on data technologies and communications with aim to promote the bridging from fundamental research on data science and networking to data engineering and communications that lead to industry products, business knowledge and standardisation.

Indexed by SCOPUS, INSPEC, EI Compendex.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at <https://link.springer.com/bookseries/15362>

Sergii Babichev · Volodymyr Lytvynenko
Editors

Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making

2022 International Scientific Conference
“Intellectual Systems of Decision-Making
and Problems of Computational Intelligence”,
Proceedings

 Springer

Editors

Sergii Babichev
Jan Evangelista Purkyně University
in Ústí nad Labem
Ústí nad Labem, Czech Republic

Volodymyr Lytvynenko
Kherson National Technical University
Kherson, Ukraine

Kherson State University
Kherson, Ukraine

ISSN 2367-4512

ISSN 2367-4520 (electronic)

Lecture Notes on Data Engineering and Communications Technologies

ISBN 978-3-031-16202-2

ISBN 978-3-031-16203-9 (eBook)

<https://doi.org/10.1007/978-3-031-16203-9>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2023, corrected publication 2023

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Data engineering, collecting, analyzing and processing information are the current directions of modern computer science. Many areas of current existence generate a wealth of information which should be stored in a structured manner, analyzed and processed appropriately in order to gain the knowledge concerning investigated process or object. Creating new modern information and computer technologies for data analysis and processing in various fields of data mining and machine learning creates the conditions for increasing the effectiveness of the information processing by both the decrease of time and the increase of accuracy of the data processing.

The international scientific conference “Intellectual Decision-Making Systems and Problems of Computational Intelligence” is a series of conferences performed in East Europe. They are very important for this geographic region since the topics of the conference cover the modern directions in the field of artificial and computational intelligence, data mining, machine learning and decision making. The aim of the conference is the reflection of the most recent research in the fields of artificial and computational intelligence used for solving problems in a variety of areas of scientific research related to computational intelligence, data mining, machine learning and decision making.

The current ISDMCI’2022 Conference held in Rivne, Ukraine, from June 14 to 16, 2022, was a continuation of the highly successful ISDMCI conference series started in 2006. For many years, ISDMCI has been attracting hundreds or even thousands of researchers and professionals working in the field of artificial intelligence and decision making. This volume consists of 39 carefully selected papers that are assigned to three thematic sections:

Section 1. Analysis and Modeling of Hybrid Systems and Processes:

- Methods and tools of system modeling under uncertainty
- Problems of identification of hybrid system, models and processes
- Modeling of the operating hybrid systems
- Modeling of dynamic objects of various nature
- Time series forecasting and modeling
- Information technology in education

Section 2. Theoretical and Applied Aspects of Decision-Making Systems:

- Decision-making methods
- Multicriterial models of decision-making under uncertainty
- Expert systems of decision-making
- Methods of artificial intelligence in decision-making systems
- Software and tools for synthesis of decision-making systems
- Applied systems of decision-making support

Section 3. Data Engineering, Computational Intelligence and Inductive Modeling:

- Inductive methods of hybrid systems modeling
- Data engineering
- Computational linguistics
- Data mining
- Multiagent systems
- Neural networks and fuzzy systems
- Evolutionary algorithm and artificial immune systems
- Bayesian networks
- Fractals and problems of synergetics
- Images recognition, cluster analysis and classification models

We hope that the broad scope of topics related to the fields of artificial intelligence and decision making covered in this proceedings volume will help the reader to understand that the methods of computational intelligence, data mining and machine learning are important elements of modern computer science.

June 2022

Oleh Mashkov
Yuri Krak
Sergii Babichev
Viktor Moshynskiy
Volodymyr Lytvynenko

Organization

ISDMCI'2020 is organized by the Department of Informatics and Computer Science, Kherson National Technical University, Ukraine, in cooperation with:

National University of Water and Environmental Engineering, Ukraine

Kherson State University, Kherson, Ukraine

Jan Evangelista Purkyně University in Ústí nad Labem, Czech Republic

Uzhhorod National University, Ukraine

Lublin University of Technology, Poland

Taras Shevchenko National University, Ukraine

V. M. Glushkov Institute of Cybernetics NASU, Ukraine

Program Committee

Chairman

Oleh Mashkov

State Ecological Academy of Postgraduate
Education and Natural Resources
Management of Ukraine, Kyiv, Ukraine

Vice-chairmen

Yuri Krak

Taras Shevchenko National University, Kyiv,
Ukraine

Sergii Babichev

Jan Evangelista Purkyně University in Ústí nad
Labem, Ústí nad Labem, Czech Republic;
Kherson State University, Kherson, Ukraine

Volodymyr Lytvynenko

Kherson National Technical University, Kherson,
Ukraine

Members

Natalia Axak	Kharkiv National University of Radio Electronics, Ukraine
Tetiana Aksenova	Grenoble University, France
Svitlana Antoshchuk	Odessa National Polytechnic University, Ukraine
Olena Arsirii	Odessa National Polytechnic University, Ukraine
Sergii Babichev	Jan Evangelista Purkyně University in Ústí nad Labem, Czech Republic; Kherson State University, Ukraine
Alexander Barmak	Khmelnitsky National University, Ukraine
Vitor Basto-Fernandes	University Institute of Lisbon, Portugal
Juri Belikov	Tallinn University of Technology, Estonia
Andrii Berko	Lviv Polytechnic National University, Ukraine
Oleg Berezkiy	Ternopil National Economic University, Ukraine
Oleg Bisikalo	Vinnitsia National Technical University, Ukraine
Peter Bidyuk	National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine
Oksana Bihun	Mathematics University of Colorado, Colorado Springs, USA
Yevgeniy Bodyanskiy	Kharkiv National University of Radio Electronics, Ukraine
Yevheniy Burov	Lviv Polytechnic National University, Ukraine
Zoran Cekerevac	“Union – Nikola Tesla” University, Serbia
Sergiu Cataranciuc	Moldova State University, Moldova Republic
Ibraim Didmanidze	Batumi Shota Rustaveli State University, Georgia
Michael Emmerich	Leiden Institute of Advanced Computer Science, Leiden University, the Netherlands
Oleg Garasym	Volvo IT, Poland
Fedir Geche	Uzhhorod National University, Ukraine
Sergiy Gnatyuk	National Aviation University, Ukraine
Vladimir Golovko	Brest State Technical University, Belarus
Oleksii Gorokhovatskyi	Simon Kuznets Kharkiv National University of Economics, Ukraine
Aleksandr Gozhij	Petro Mohyla Black Sea National University, Ukraine
Natalia Grabar	CNRS UMR 8163 STL, France
Klaus ten Hagen	Zittau Görlitz University of Applied Sciences, Germany
Volodymyr Hnatushenko	Dnipro University of Technology, Ukraine
Viktorya Hnatushenko	National Metallurgical Academy of Ukraine, Ukraine
Volodymyr Hrytsyk	Lviv Polytechnic National University, Ukraine
Ivan Izonin	Lviv Polytechnic National University, Ukraine

Irina Ivashenko	Karpenko Physico-Mechanical Institute of the NAS of Ukraine, Ukraine
Irina Kalimina	Petro Mohyla Black Sea National University, Ukraine
Maksat Kalimoldayev	Institute of Information and Computational Technologies, Kazakhstan
Viktor Kaplun	Kyiv National University of Technologies and Design, Ukraine
Bekir Karlik	Neurosurgical Simulation Research and Training Centre, Canada
Alexandr Khimich	Glushkov Institute of Cybernetic of NAS of Ukraine, Ukraine
Lyudmyla Kirichenko	Kharkiv National University of Radio Electronics, Ukraine
Pawel Komada	Lublin University of Technology, Poland
Konrad Gromaszek	Lublin University of Technology, Poland
Pavel Kordik	Czech Technical University in Prague, Czech Republic
Mykola Korablyov	Kharkiv National University of Radio Electronics, Ukraine
Andrzej Kotyra	Lublin University of Technology, Poland
Yuri Krak	Taras Shevchenko National University, Ukraine
Jan Krejci	Jan Evangelista Purkyně University in Ústí nad Labem, Czech Republic
Evelin Krmac	University of Ljubljana, Slovenia
Roman Kuc	Yale University, Yale, USA
Evgeniy Lavrov	Sumy State University, Ukraine
Frank Lemke	Knowledge Miner Software, Germany
Igor Liakh	Uzhhorod National University, Ukraine
Volodymyr Lytvynenko	Kherson National Technical University, Ukraine
Vasyl Lytvyn	Lviv Polytechnic National University, Ukraine
Leonid Lyubchik	National Technical University “Kharkiv Polytechnic Institute”, Ukraine
Igor Malets	Lviv State University of Life Safety, Ukraine
Viktor Morozov	Taras Shevchenko National University, Ukraine
Viktor Moshynskiy	National University of Water and Environmental Engineering, Ukraine
Viktor Mashkov	Jan Evangelista Purkyně University in Ústí nad Labem, Czech Republic
Mykola Malyar	Uzhhorod National University, Ukraine
Sergii Mashtalir	Kharkiv National University of Radio Electronics, Ukraine
Jiří Škvor	Jan Evangelista Purkyně University in Ústí nad Labem, Czech Republic

Jiří Fišer	Jan Evangelista Purkyně University in Ústí nad Labem, Czech Republic
Sergii Olszewski	Taras Shevchenko National University, Ukraine
Opeyemi Olakitan	Cornell University, UK
Volodymyr Osypenko	Kyiv National University of Technologies and Design, Ukraine
Sergii Pavlov	Vinnytsia National Technical University, Ukraine
Nataliya Pankratova	National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine
Anatolii Pashko	Taras Shevchenko National University of Kyiv, Ukraine
Dmytro Peleshko	GeoGuard, Ukraine
Iryna Perova	Kharkiv National University of Radio Electronics, Ukraine
Eduard Petlenkov	Tallinn University of Technology, Estonia
Volodymyr Polishchuk	Uzhhorod National University, Ukraine
Michael Pokojov	Karlsruher Institut für Technologie (KIT), Universität Konstanz, Mannheim Area, Germany
Taras Rak	IT Step University, Ukraine
Yuriy Rashkevych	Lviv National Polytechnic University, Ukraine
Hanna Rudakova	Kherson National Technical University, Ukraine
Yuriy Romanyshyn	Lviv Polytechnic National University, Ukraine
Yuri Samokhvalov	Taras Shevchenko National University, Ukraine
Silakari Sanjay	Rajiv Gandhi Technical University, Madhya Pradesh, India
Andrii Safonyk	National University of Water and Environmental Engineering, Ukraine
Natalia Savina	National University of Water and Environmental Engineering, Ukraine
Antonina Savka	Openet, Ireland
Galina Setlak	Rzeszow University of Technology, Poland
Natalya Shakhovska	Lviv Polytechnic National University, Ukraine
Manik Sharma	DAV University, India
Volodimir Sherstyuk	Kherson National Technical University, Ukraine
Galyna Shcherbakova	Odessa National Polytechnic University, Ukraine
Juergen Sieck	Humboldt-Universität zu Berlin, Germany
Miki Sirola	Institute for Energy Technology, Norway
Andrzej Smolarz	Lublin University of Technology, Poland
Marian Sorin Nistor	Bundeswehr University Munich, Germany
Vasyl Teslyuk	Lviv Polytechnic National University, Ukraine
Roman Tkachenko	Lviv Polytechnic National University, Ukraine
Vasyl Trysnyuk	Institute of Telecommunications and Global Information Space, Ukraine
Ivan Tsmots	Lviv Polytechnic National University, Ukraine

Oleksii Tyshchenko	Institute for Research and Applications of Fuzzy Modeling, CEIT Innovations, University of Ostrava, Czech Republic
Oleksandr Trofymchuk	Institute of Telecommunications and Global Information Space, Ukraine
Kristina Vassiljeva	Tallinn University of Technology, Estonia
Viktor Voloshyn	IT Step University, Ukraine
Olena Vynokurova	GeoGuard, Ukraine
Victoria Vysotska	Lviv Polytechnic National University, Ukraine
Waldemar Wojcik	Lublin University of Technology, Poland
Stefan Wolfgang Pickl	Bundeswehr University Munich, Germany
Mykhaylo Yatsymirskyy	Institute of Information Technology, Lodz University of Technology, Poland
Sergey Yakovlev	National Aerospace University “Kharkiv Aviation Institute”, Ukraine
Iryna Evseyeva	University of Newcastle, England
Danuta Zakrzewska	Institute of Information Technology, Lodz University of Technology, Poland
Elena Zaitseva	Zilinska Univerzita v Ziline, Slovakia
Maryna Zharikova	Kherson National Technical University, Ukraine

Organization Committee

Chairman

Viktor Moshynskiy	National University of Water and Environmental Engineering, Ukraine
-------------------	---

Vice-chairmen

Volodymyr Lytvynenko	Kherson National Technical University, Ukraine
Natalia Savina	National University of Water and Environmental Engineering, Ukraine

Members

Igor Baklan	National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine
Oleg Boskin	Kherson National Technical University, Ukraine
Liliya Chyrun	Polytechnic National University, Ukraine
Nataliya Kornilovska	Kherson National Technical University, Ukraine
Yurii Lebedenko	Kherson National Technical University, Ukraine
Irina Lurje	Kherson National Technical University, Ukraine
Oksana Ohnieva	Kherson National Technical University, Ukraine
Viktor Peredery	Kherson National Technical University, Ukraine

Oleg Riznyk
Polina Zhernova

Lviv Polytechnic National University, Ukraine
Kharkiv National University of Radio
Electronics, Ukraine

Svetlana Vyshemyrskaya
Mariia Voronenko

Kherson National Technical University, Ukraine
Kherson National Technical University, Ukraine

Contents

Analysis and Modeling of Hybrid Systems and Processes	
Application of Convolutional Neural Network for Gene Expression Data Classification	3
Lyudmyla Yasinska-Damri, Sergii Babichev, Bohdan Durnyak, and Tatiana Goncharenko	
Formation of Subsets of Co-expressed Gene Expression Profiles Based on Joint Use of Fuzzy Inference System, Statistical Criteria and Shannon Entropy	25
Igor Liakh, Sergii Babichev, Bohdan Durnyak, and Iryna Gado	
Mathematical Model of Preparing Process of Bulk Cargo for Transportation by Vessel	42
Oksana Polyvoda and Vladyslav Polyvoda	
Computer Simulation of Joule-Thomson Effect Based on the Use of Real Gases	61
Vasiliy Nadraga, Anatolii Balanda, Mariana Polodiuk, Yuliia Bobyr, and Tetiana Kochura	
Simulating Soil Organic Carbon Turnover with a Layered Model and Improved Moisture and Temperature Impacts	74
Olha Stepanchenko, Liubov Shostak, Viktor Moshynskiy, Olena Kozhushko, and Petro Martyniuk	
Optimization of Coagulant Dosing Process for Water Purification Based on Artificial Neural Networks	92
Andrii Safonyk and Myroslav Matviichuk	
Methodology for Solving Forecasting Problems Based on Machine Learning Methods	105
Irina Kalinina and Aleksandr Gozhyj	

The Comprehensive Model of Using In-Depth Consolidated Multimodal Learning to Study Trading Strategies in the Securities Market	126
Nataliya Boyko	
Mathematical and Computer Model of the Tree Crown Ignition Process from a Mobile Grassroots Fire	148
Oksana Karabyn, Olga Smotr, Andrij Kuzyk, Igor Malets, and Vasy Karabyn	
Features of Complex Application of the Formal Method of EVENT-B for Development of Environmental Management Systems	160
Oleh Mashkov, Oleh Ilyin, Viktor Mashkov, Oleh Boskin, and Oksana Ohnieva	
Ecology Objects Recognition by Optimized Inverse Filtration of Textured Background	177
Roman Kvyetnyy, Olga Sofina, and Yuriy Bunyak	
Theoretical and Applied Aspects of Decision-Making Systems	
Information Technology to Assess the Enterprises' Readiness for Innovative Transformations Using Markov Chains	197
Marharyta Sharko, Olha Liubchuk, Galina Krapivina, Natalia Petrushenko, Olga Gonchar, Kateryna Vorobyova, and Nataliia Vasylenko	
Method to Find the Original Source of COVID-19 by Genome Sequence and Probability of Electron Capture	214
Yoshio Matsuki, Aleksandr Gozhyj, Irina Kalinina, and Peter Bidyuk	
Leader-Follower Strategy of Fixed-Wing Unmanned Aerial Vehicles via Split Rejoin Maneuvers	231
Roneel Chand, Krishna Raghuwaiya, and Jito Vanualailai	
Prognostic Assessment of COVID-19 Vaccination Levels	246
Iryna Pikh, Vsevolod Senkivskyy, Alona Kudriashova, and Nataliia Senkivska	
Application of the Theory of Functional Stability in the Problems of Covering Territories by Sensory Networks	266
Oleh Mashkov, Alexey Bychkov, Ganna Kalahnik, Victor Shevchenko, and Svitlana Vyshemyrska	
Adaptive Decision-Making Strategies in the Game with Environment	286
Petro Kravets, Victoria Vysotska, Vasyl Lytvyn, and Lyubomyr Chyrun	

System Analysis of the Internal and External Migration Processes in Ukraine 302
 Andrii Roskladka, Nataliia Roskladka, Olexander Romanyuk, Tetiana Troianovska-Korobeinikova, and Liudmyla Savytska

Associative Information Retrieval in Medical Databases 320
 Anatoliy Povoroznyuk, Anna Filatova, Oksana Povoroznyuk, and Iryna Shakhina

Analysis of Deep Learning Methods in Adaptation to the Small Data Problem Solving 333
 Iurii Krak, Vladyslav Kuznetsov, Serhii Kondratiuk, Larisa Azarova, Olexander Barmak, and Pavlo Padiuk

Cognitive and Information Decision Support Technologies for Operational Management of Energy-Active Objects in Boundary Modes. 353
 Lubomyr Sikora, Natalya Lysa, Roman Martysyshyn, and Yuliya Miyushkovych

Expert Decision Support System Modeling in Lifecycle Management of Specialized Software. 367
 Yuliia Kordunova, Oleksandr Prydatko, Olga Smotr, and Roman Golovaty

Data Engineering, Computational Intelligence and Inductive Modeling

Machine Learning of the Biotechnic System for Gastroesophageal Reflux Disease Monitoring 387
 Vsevolod Novikov, Mariia Voronenko, Anastasiia Novikova, Oleg Boskin, Oleksii Tyshchenko, Yuriy Rozov, Yuriy Bardachov, and Svitlana Vyshemyrska

Processing Technology of Thematic Identification and Classification of Objects in the Multispectral Remote Sensing Imagery 407
 Volodymyr Hnatushenko, Yana Shedlovska, and Igor Shedlovsky

CTrace: Language for Definition of Epidemiological Models with Contact-Tracing Transmission 426
 Vladyslav Sarnatskyi and Igor Baklan

Optimization of Data Preprocessing Procedure in the Systems of High Dimensional Data Clustering 449
 Maksym Korobchynskyi, Myhailo Rudenko, Vladyslav Dereko, Oleksandr Kovtun, and Oleksandr Zaitsev

Features of the Application of the Principal Component Method to the Study of Acoustic Emission Signals Under Loading of Multilayer Structures 462
Petr Louda, Oleksandr Sharko, Dmitry Stepanchikov, and Artem Sharko

Computational Intelligence in Medicine 488
Oleh Berezsky, Oleh Pitsun, Petro Liashchynskyi, Bohdan Derysh, and Natalia Batryn

Approaches and Techniques to Improve Machine Learning Performance in Distributed Transducer Networks 511
Mykola Hodovychenko, Svitlana Antoshchuk, Ivan Lobachev, Thorsten Schöler, and Mykhaylo Lobachev

Investigation of the Impact of Primary Data Processing on the Results of Neural Network Training for Satellite Imagery Recognition 525
Dmytro Soldatenko and Viktoriia Hnatushenko

Application of Wavelet Transform for Machine Learning Classification of Time Series 547
Lyudmyla Kirichenko, Oksana Pichugina, Tamara Radivilova, and Kyrylo Pavlenko

A Noise Resistant Credibilistic Fuzzy Clustering Algorithm on a Unit Hypersphere with Illustrations Using Expression Data 564
Zhengbing Hu, Mark Last, Tzung-Pei Hong, Oleksii K. Tyshchenko, and Esha Kashyap

Visual Analytics-Based Method for Sentiment Analysis of COVID-19 Ukrainian Tweets 591
Oleksii Kovalchuk, Vitalii Slobodzian, Olena Sobko, Maryna Molchanova, Olexander Mazurets, Oleksander Barmak, Iurii Krak, and Nataliia Savina

Software Based on Ontological Tasks Models 608
Yevhen Burov, Victoria Vysotska, Vasyl Lytvyn, and Lyubomyr Chyrun

Neural Network Analysis of Evacuation Flows According to Video Surveillance Cameras 639
Oleksandr Khlevnoi, Nazarii Burak, Yurii Borzov, and Diana Raita

Real-Time Information Technology Human Detection Using Cloud Services 651
Natalya Sokolova, Yuliia Zhuravlova, Oleksandr Mushtat, and Yevhen Obydennyi

Deep Learning Technology for Automatic Burned Area Extraction Using Satellite High Spatial Resolution Images 664
Vita Kashtan and Volodymyr Hnatushenko

Classification Methods of Heterogeneous Data in Intellectual Systems of Medical and Social Monitoring 686
Olena Arsirii, Svitlana Antoshchuk, Olga Manikaeva, Oksana Babilunha, and Anatolii Nikolenko

IaaS-Application Development for Paralleled Remote Sensing Data Stream Processing 705
Vadym Zhernovyi, Volodymyr Hnatushenko, and Olga Shevtsova

Correction to: Classification Methods of Heterogeneous Data in Intellectual Systems of Medical and Social Monitoring C1
Olena Arsirii, Svitlana Antoshchuk, Olga Manikaeva, Oksana Babilunha, and Anatolii Nikolenko

Author Index 719

Analysis and Modeling of Hybrid Systems and Processes



Application of Convolutional Neural Network for Gene Expression Data Classification

Lyudmyla Yasinska-Damri¹ , Sergii Babichev^{2,3}  , Bohdan Durnyak¹ ,
and Tatiana Goncharenko³ 

¹ Ukrainian Academy of Printing, Lviv, Ukraine
durnyak@uad.lviv.ua

² Jan Evangelista Purkyně University in Ústí nad Labem,
Ústí nad Labem, Czech Republic
sergii.babichev@ujep.cz

³ Kherson State University, Kherson, Ukraine
{sbabichev, TGoncharenko}@ksu.ks.ua

Abstract. The results of research regarding the development of a gene expression data classification system based on a convolutional neural network are presented. Gene expression data of patients who were studied for lung cancer were used as experimental data. 156 patients were studied, of which 65 were identified as healthy and 91 patients were diagnosed with cancer. Each of the DNA microchips contained 54,675 genes. Data processing was carried out in two stages. In the first stage, 10,000 of the most informative genes in terms of statistical criteria and Shannon entropy were allocated. In the second stage, the classification of objects containing as attributes the expression of the allocated genes was performed by using a convolutional neural network. The obtained diagrams of the data classification accuracy during both the neural network model learning and validation indicate the absence of the network retraining since the character of changing the accuracy and loss values on trained and validated subsets during the network learning procedure implementation is the same within the allowed error. The analysis of the obtained results has shown, that the accuracy of the object's classification on the test data subset reached 97%. Only one object of 39 was identified incorrectly. This fact indicates the high efficiency of the proposed model.

Keywords: Gene expression data · Classification · Convolution neural network · Classification quality criteria

1 Introduction

The development of hybrid models to form the subsets of differentially expressed and mutually correlated gene expression profiles based on the joint application of both data mining methods and machine learning tools is one of the current

directions of modern bioinformatics. Differentially expressed genes in this case means genes whose level of expression differs as much as possible for different types of objects. In other words, the values of the same genes expression for sick and healthy patients or for sick patients with different disease states should be as possible differ. On the other hand, gene expression profiles should be as mutually correlated as possible. In the presence of such a subset of differentially expressed and mutually correlated gene expression profiles creates the conditions for the development of highly effective systems for diagnosing patients' health based on gene expression data that can identify early diseases such as various types of cancer, Parkinson's disease, Alzheimer's disease, neuroblastoma, etc. The formation of such subsets of gene expression profiles creates conditions for the reconstruction of highly informative gene regulatory networks, analysis of simulation results of which will better understand the interaction between target genes and transcription factors to form appropriate actions to change the expression of relevant genes and/or transcription factors to neutralize appropriate disease by using modern effective drugs and/or methods of treatment.

It should be noted that traditional algorithms of data mining and machine learning are not effective for gene expression profiles, which are characterized by a large number of attributes (high dimensionality). Thus, density clustering algorithms [21, 22], which allow us with high precision to allocate complex clusters of objects with a small number of attributes, are not effective for gene expression profiles. Therefore, there is a problem of choosing or developing a clustering algorithm or an ensemble of algorithms that focus on high-dimensional gene expression profiles and take into account the principles of objective clustering inductive technology [3, 4]. On the other hand, an objective assessment of the adequacy of the clustering algorithm to form subsets of gene expression profiles is possible by analyzing the results of the classification of objects that contain the values of the expressions of selected genes as attributes. In [3], the authors present the results of research regarding the use of the Random Forest algorithm as a classifier, which has shown high efficiency for the classification of gene expression profiles when the two classes of objects were contained in the dataset. For this reason, there is a necessity for comparative analysis of various types of classifiers, including neural networks, in order to create a classifier focused on high-dimensional gene expression data, which can allow us to identify objects with high accuracy in multiclass tasks. As a result, the conditions are created for the development of a hybrid model of forming subsets of differentially expressed and mutually correlated gene expression profiles, in which cluster analysis is used to group profiles, and clustering adequacy is assessed by using a classifier focused on this data type. Within the framework of our research, we evaluate the efficiency of a convolutional neural network (CNN) to classify the objects, the attributes of which are high dimensional gene expressions.

2 Literature Review

A large number of scientific papers are currently devoted to solving the problem of using convolutional neural networks (CNN) for the classification of large

amounts of data [7–9, 12]. So, in paper [9], the authors have collected 6136 human samples from 11 types of cancer disease with the following transformation of the obtained gene expressions datasets into 2D images by application of spectral clustering technique. In the next step, they have applied the convolutional neural network for the obtained image classification. The classification accuracy was changed within the range from 98.4% to 97.7% during the simulation procedure implementation. The paper [7] devoted to the application of both the support vector machine algorithm and convolutional neural network to identify the early signs of breast cancer. The authors have shown that the proposed technique outperforms the current methods for mass regions classification into benign and malignant. To the authors' mind, this fact can help radiologists reduce their burden and enhance their sensitivity when the prediction of malignant masses. The results of the research regarding the creation of the technique of a dense skip connection encoding-decoding based on a deep convolutional neural network are presented in [12]. In this work, the authors have proposed the image preprocessing method to provide much more texture information and enhance the contrast between thymoma and its surrounding tissues. They have shown that an application of the proposed technique allows increasing the objects classification accuracy by 4% in comparison with other well-known methods. In [8], the authors to solve the problem of data imbalance have proposed a Mixed Skin Lesion Picture Generate method based on Mask R-CNN. They designed a melanoma detection framework of Mask-DenseNet+ based on MSLP-MR. This method used Mask R-CNN to introduce the method of mask segmentation and combined with the idea of ensemble learning to integrate multiple classifiers for weighted prediction. They also have shown that the experimental results on the ISIC dataset have allowed them to obtain the accuracy of the algorithm is 90.61%, the sensitivity reaches 78.00%, which is higher than the well known original methods. In [16, 18], the authors presented the results of research concerning the use of CNN in models of object classification based on gene expression data. The analysis of the obtained results has shown the high efficiency of CNN to classify the objects that contain large amounts of attributes.

However, it should be noted, that despite some advances in this subject area, the problem of creating an optimal model of CNN for the classification of objects based on gene expression data has no currently unambiguous solution. The main problem consists of the absence of effective technique for determining CNN hyperparameters and forming the network structure considering the type of the investigated data. This problem can be solved by applying the modern effective data science techniques which are used successfully in various fields of the scientific research nowadays [13, 14, 19].

3 Material and Methods

3.1 Architecture, Structure and Model of Convolutional Neural Network

The general architecture of CNN is shown in Fig. 1. As an input layer, in this case, it can be used the vector of values of gene expressions of one of the studied

samples. The number of genes can reach several thousand, so the use of classical neural networks is complicated by a large number of neurons, which complicates the choice of optimal network architecture. Images, sound signals, time series, electrocardiograms, etc. can be used as input data when using CNN.

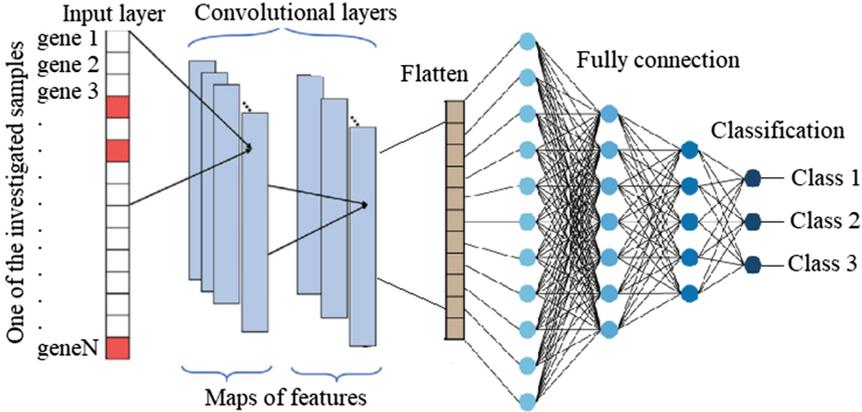


Fig. 1. The general architecture of convolutional neural network

As it can be seen from Fig. 1, the first stage of data processing in CNN is the convolution operation. This operation is one of the most important and involves the decomposition of input data on feature maps by applying the appropriate convolution kernel. Each feature map, in this instance, is presented as a tensor of parameters corresponding to a limited part of the investigated data. Thus, the first step is to some extent a filter used to decompose the input dataset into components, each of which contains information about the corresponding data component. For this reason, the choice of the number of filters (feature maps) is one of the most important steps in the stage of initialization of network parameters and it is one of the hyperparameters of the network. Increasing the number of filters leads to an increase in the level of the data features detail, but too many features can lead to the retraining of the network. In other words, can be identified the groups of features that are not characteristic of the studied object. This fact can lead to a large discrepancy in the accuracy of the classification of objects during the network learning and its further testing and validation of the obtained model. On the other hand, a small number of filters can reduce the resolution of the network. Some significant components of the signal that have significant weight in the classification of the object will not be allocated. This fact will also worsen the results of classification at both learning and model testing and validation.

The mathematical model of the convolution operation of the input vector of gene expression values with the formation of the corresponding map of features can be represented as follows:

$$s(t) = (e * w)(t) = \int_{-\infty}^{+\infty} e(\tau)w(t - \tau)d\tau \quad (1)$$

where: $s(t)$ is the map of t -th group of features; e is the appropriate gene expression value; w is the kernel function or a features detector for forming a group of mutually correlated features of a gene expression vector.

The implementation of the convolution operation also assumes that the value of the kernel function outside the corresponding group of features will be zero, so the infinite sum used in formula (1) can be replaced by finite sums with values in the selected range for the corresponding feature map. The main hyperparameters of the function of the convolution operation implementation should also include the step of bypassing the signal s . It is obvious that increasing the value of this parameter leads to a decrease in the number of feature maps. In other words, the model, in this case, is simpler, but the accuracy of its operation may decrease due to the imposition of signal characteristics for different fields. The next hyperparameter of the model is the padding by zeros p . The use of this parameter allows us to preserve the dimensionality of the input data when forming feature maps, while values that do not correspond to the selected features are supplemented by zeros. The dimensionality of feature filters is determined by the fact that increasing the dimensionality will lead to the selection of more features, and additional selected features are usually uncommon, but they may have more weight to identify the studied object. The general process of transformation of part of the gene expression vector into the corresponding convolutional layer can be represented by the following equation:

$$y = f(w * e + b) \quad (2)$$

where: y is a map of features formed on the corresponding convolutional layer; w - kernel function or feature detector; e - vector of gene expression values (input data vector); b is a parameter that determines the step of the filter shift during the convolution operation implementing.

The next stage in the CNN model implementation is the aggregation of the selected convolutional layers (flatten) in order to form a fully connected layer at the next step of this procedure. As was noted hereinbefore, one of the advantages of the convolution layer is that the convolution is a local operation. The spatial length of the kernel is usually small. One feature map element is calculated using only a small number of elements at its input. The necessity of forming a fully connected layer is determined by the fact that to calculate any element at the output of the network (based on the features of the object which should be identified when solving the classification problem) requires all the elements at the input of the classifier. Thus, the procedure of the layers flattening to form a fully connected layer allows us to use the full vector or matrix of input data to learn the depth model of CNN, while at the stage of forming the feature maps, was performed the stage of data filtering to identify the most informative features.

As it can be seen from Fig. 1, at an early stage of a fully connected layer formation, it is necessary to form the pooling layers, which can be global and

contain the full vector of input signal attributes or local, and contain only the vector of the input signal attributes subset. These layers will be applied to the corresponding neurons of the current layer. The main purpose of the flattening step is to form a vector of data attributes of smaller size compared to the dimension of the input data while maintaining the most informative characteristics of the input signal by forming connections between the corresponding neurons from the previous layer and one neuron of the current layer. For this reason, the flattening operation is usually applied step by step between the formed convolutional layers [20]. Flatten can be performed by averaging, maximizing or minimizing operations, with the average value calculated for the corresponding group of neurons in the first case, the maximum or minimum in the second and third cases, respectively.

Formally, the listed hereinbefore operations of different types of flattening can be represented as follows:

– averaging:

$$e(i, j) = \frac{\sum_{s,p} x(i-s, j-p)}{s * p} \quad (3)$$

– maximizing:

$$e(i, j) = \max_{s,p} (x(i-s, j-p)) \quad (4)$$

– minimizing:

$$e(i, j) = \min_{s,p} (x(i-s, j-p)) \quad (5)$$

where: $e(i, j)$ is the expression value of the current level element, the position of which is determined by the coordinates i and j ; x is the expressions value of the element at previous levels with coordinates $(i-s, j-p)$; s, p are the dimension of the field of the corresponding features.

It should be noted that the result of the flattening layers formation is determined by similar hyperparameters that determine the result of the convolution operation. Usually, the signal bypass step b and the dimension of the corresponding receptive field are used as hyperparameters. The architecture of a fully connected layer in CNN usually corresponds to the architecture of ordinary neural networks, in which all neurons of the current layer are connected to neurons of the previous layer. As can be seen in Fig. 1, the full-connected layer in CNN is applied in the penultimate step after the stages of convolution and flattening which allows us to significantly simplify the network architecture of the full-connected layer by reducing the dimension of the input data vector. The values of neurons' weights of the corresponding fully connected level are determined by a similar formula, according to which the convolution layers are formed with the difference that w is an array (or vector) of parameters of the corresponding fully connected layer:

$$y = f(w^T * e + b) \quad (6)$$

Similar to classical neural networks, neurons of the fully connected layer of CNN involve the use of the activation function to calculate the value of the

neurons' output based on the vector of input data. The following functions are usually used as activation functions of CNN output layer neurons:

- sigmoidal function:

$$y = \frac{1}{1 + \exp(-e)} \quad (7)$$

where e determines the value of the expression supplied to the input of the neuron. The value of y , in this case, varies within the range $[0,1]$, which makes it convenient to use the sigmoidal function at the last level of the fully connected layer in the classifiers;

- softmax function:

$$y = \frac{\max_{i=1,N} \exp(e_i)}{\sum_{j=1}^N \exp(e_j)} \quad (8)$$

where: e_i is the expression values of the i -th element of the previous layer; N is the number of elements on the previous layer.

The use of this function also leads to the normalization of the input data within the range $[0,1]$, however, if the application of the sigmoidal function is appropriate mainly for solving binary classification problems, the Softmax function usually allows us to obtain high accuracy classification when the multiclass dataset is applied and one or another class is determined by the maximum probability at the output of the neuron of the output layer. In addition, this function is more stable to noise in comparison with the sigmoidal function. Increasing the values of neurons' output at the previous level to a constant value almost does not change the value at the output of the neuron of the current layer. As was noted hereinbefore, sigmoidal or Softmax activation functions are usually used at the last fully connected layer. Other activation functions such as linear, tangential, hyperbolic tangent, etc., are less effective for this layer. For neurons of the input and middle layers, the most common is the linear activation function, but the choice of the appropriate activation function usually is determined empirically at the simulation stage, taking into account the type of investigated data. It should be noted that the choice of activation functions for neurons of the respective layers significantly affects the performance of the network because if you choose the wrong combination of activation functions, the correct setting of network parameters in the learning process can be problematic.

The control of network training, in order to adjust the neurons' parameters, is performed by calculating the loss values at the output of the network. In this instance, the comparison of the value at the output of the network with the appropriate value corresponding to the current signal supplied to the input of the network. The loss function, in this case, is one of the main criteria when configuring network parameters is carried out, since it largely determines the effectiveness of the network learning procedure. The main types of loss functions include the cross-entropy function and the mean-squared error function [5,6]. When using CNN, at the output of the network the probability that the current data vector (in our case, gene expression) belongs to one or another class is calculated. By comparing the current probabilities of belonging of the data vector

to a particular class with the target probabilities (0 or 1) in accordance with the principle of plausibility, the network parameters are adjusted so as to minimize differences in network output. In this case, network training is usually carried out by applying a standard algorithm of back error propagation. The cross-entropy function is calculated in this instance as follows [11]:

$$L(e, p) = - \sum_{i=1}^N t_i \cdot \log_2(p_i) \quad (9)$$

where: e is the current vector of gene expression values that determine the state of the studied object; N is the number of classes to which the object may belong; t_i is the target i -th class to which the current vector of gene expression values belongs; p_i is the probability that the current vector of gene expression values belongs to the i -th class, which is calculated by CNN.

It should be noted that the probability of network output can be calculated by using any activation function, but usually, for a wide range of tasks, the Softmax function is used in the case of both binary and multiclass classifications. The sigmoidal activation function can also be used for binary classification. As mentioned earlier, the choice of the activation function is determined by the structure of the data and in each case is carried out separately by evaluating the simulation results.

4 Simulation, Results and Discussion

4.1 Gene Expression Dataset Formation and Preprocessing

Simulation of the convolutional neural network and evaluation of its effectiveness for the gene expression data classification was performed using GSE19188 gene expression dataset [17] of patients studied for lung cancer. These data were taken from the GSE (Gene Expression Omnibus) database [2] and contain the results of DNA analysis of 156 patients using DNA microchip technology, where 65 patients were identified as healthy and 91 patients were diagnosed with cancer. Each microchip contained 54675 genes. Gene expression matrices were formed by using the function of the Bioconductor package [1] of the R programming language [15]. In the first stage, a reduction operation was applied to gene expression profiles (gene expression vectors defined for all samples (vector length 156)) to remove uninformative genes by statistical criteria and Shannon entropy. We have used, as statistical criteria, the average of absolute values of gene expression and the variance of expression values in the respective profiles. Shannon entropy was calculated by the use of James-Stein estimator [10]. It was assumed that if the absolute value of gene expression for all studied samples and the value of the variance of the expression profile of this gene is less, and the Shannon entropy value of this profile is greater than the appropriate boundary values, this profile is removed from the database as uninformative. This condition was determined by the fact that the functioning of a biological organism is determined by about

25000 genes. Thus, approximately half of the 54675 genes are unexpressed and can be deleted without losing useful information about the health of the biological organism. Of the remaining half, some genes have low expression values for all samples, these genes are not crucial for the disease identification and they can also be deleted without significant loss of information. Moreover, if the variance of the respective gene expression profiles is small, these genes are not informative in terms of the resolution of the studied objects. The high value of Shannon entropy also corresponds to the chaotic change in gene expression (noise). This criterion can also be used to identify non-informative genes. Thus, the condition for the formation of a subset of informative genes by removing non-informative (reduction) ones, in this case, can be presented as follows:

$$\{e_{ij}\} = \left\{ \begin{array}{l} \max_{i=1, \dots, n}(e_{ij} \geq e_{bound}), \text{ and } \text{var}(e_{ij}) \geq \text{var}_{bound}, \\ \text{and } \text{entr}(e_{ij}) \leq \text{entr}_{bound} \end{array} \right\}, j = \overline{1, m} \quad (10)$$

where: n is the number of samples or objects to be examined; m is the number of genes.

The boundary values of the relevant criteria were determined empirically during the simulation process. Figure 2 shows the box plots of the criteria values distribution used to form a subset of gene expression profiles according to the formula (10).

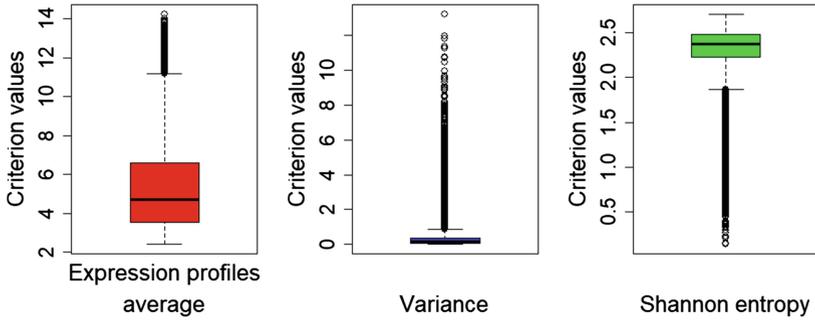


Fig. 2. Box plots of the distribution of statistical criteria and Shannon entropy values

As it can be seen from Fig. 2, most genes have relatively high expression values, and the variance and entropy of Shannon gene expression profiles are preferably high and low values respectively. This fact confirms the feasibility of forming a subset of informative gene expression profiles in accordance with the formula (10). The following threshold values of the criteria were determined as the obtained results analysis:

- absolute value of gene expressions and variance - at the level of the 25-th percentage quantile;

- Shannon entropy values – at the level of the 75-th percentage quantile.

10312 genes were allocated as the simulation result. For the convenience of further simulation procedure, the values of the criteria were slightly adjusted to allocate 10000 gene expression profiles. Thus, the dataset (156×10000) was used as experimental data during the further application of the convolutional neural network.

4.2 Application of 1D One-Layer CNN for Gene Expression Data Classification

The model of a 1D one-level convolutional neural network used to classify objects based on the formed subset of gene expression data is shown in Fig. 3.

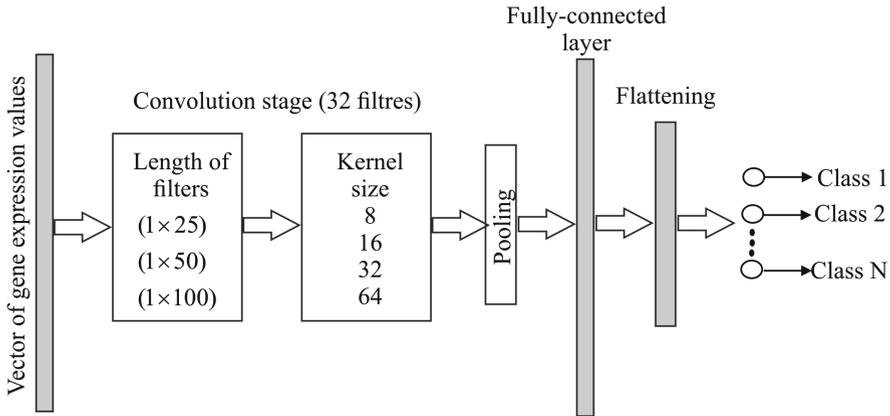


Fig. 3. Model of a 1D one-level convolutional neural network

According to the dimensionality of the experimental data, the length of the gene expression vector fed to the network input was 10000. In the first step, a filter with a length of 25, 50 and 100 genes was applied to each vector. The number of gene expression sequences that were fed step by step to the network input was 400, 200 and 100, respectively. The kernel size was 8 and 16 in the first case, 8, 16 and 32 in the second and 8, 16, 32 and 64 in the third one. The linear activation function was applied to the neurons of the convolutional layer. The kernel size of the flattening layer was 64, 128, 256 and 512. To the neurons of the flattening layer was applied the rectified linear unit function ReLU. Softmax activation function was applied to the neurons of the output layer, and the level of losses was evaluated by using the cross-entropy function.

As follows from the experimental data annotation, the examined samples can be divided into two groups (classes). The first group consists of healthy patients, and patients in the second group identified a cancer tumor. At the first stage, 156

gene expression vectors were divided into a training subset (75% - 117 samples) and a test subset (25% - 39 samples). To control the nature of the network learning process, the training network was divided into two subsets: 93 vectors of gene expression values were used directly for training and the remaining 24 for model validation. The number of learning epochs was 50. The simulation results obtained on the test subset of gene expression profiles for different combinations of hyperparameters are presented in Table 1.

Table 1. Simulation results regarding the use of 1D one-level CNN for classification of objects based on gene expression data (subset for testing)

Hyperparameters			Classification results			
Flatten layer	Filter size	Kernel size	F-index		Accuracy, %	Loss function values
			Cluster 1	Cluster 2		
64	(1,25)	8	0.96	0.98	97	0.207
64	(1,25)	16	0.9	0.94	92	0.8
128	(1,25)	8	0.9	0.94	92	0.306
128	(1,25)	16	0.96	0.98	97	0.314
256	(1,25)	8	0.88	0.94	92	0.211
256	(1,25)	16	0.87	0.92	90	0.777
512	(1,25)	8	0.9	0.94	92	0.337
512	(1,25)	16	0.9	0.94	92	0.912
64	(1,50)	8	0.96	0.98	97	0.240
64	(1,50)	16	0.93	0.96	95	0.415
64	(1,50)	32	0.96	0.98	97	0.458
128	(1,50)	8	0.87	0.92	92	0.743
128	(1,50)	16	0.87	0.92	92	0.575
128	(1,50)	32	0.90	0.94	94	0.145
256	(1,50)	8	0.96	0.98	97	0.145
256	(1,50)	16	0.93	0.96	95	0.310
256	(1,50)	32	0.96	0.98	97	0.406
512	(1,50)	8	0.92	0.96	95	0.146
512	(1,50)	16	0.96	0.98	97	0.148
512	(1,50)	32	0.96	0.98	97	0.346
64	(1,100)	8	0.96	0.98	97	0.169
64	(1,100)	16	0.93	0.96	95	0.146
64	(1,100)	32	0.89	0.93	91	0.226
64	(1,100)	64	0.90	0.94	92	0.232
128	(1,100)	8	0.92	0.96	95	0.135
128	(1,100)	16	0.96	0.98	97	0.113
128	(1,100)	32	0.96	0.98	97	0.154
128	(1,100)	64	0.93	0.96	95	0.574
256	(1,100)	8	0.96	0.98	97	0.112
256	(1,100)	16	0.96	0.98	97	0.141
256	(1,100)	32	0.96	0.98	97	0.143
256	(1,100)	64	0.92	0.96	95	0.242
512	(1,100)	8	0.96	0.98	97	0.139
512	(1,100)	16	0.93	0.96	95	0.205
512	(1,100)	32	0.90	0.94	92	0.217
512	(1,100)	64	0.93	0.96	95	0.463

Figure 4 shows the boxplots of both the investigated objects' classification quality criteria and loss function values, which were calculated during the simulation procedure implementation.

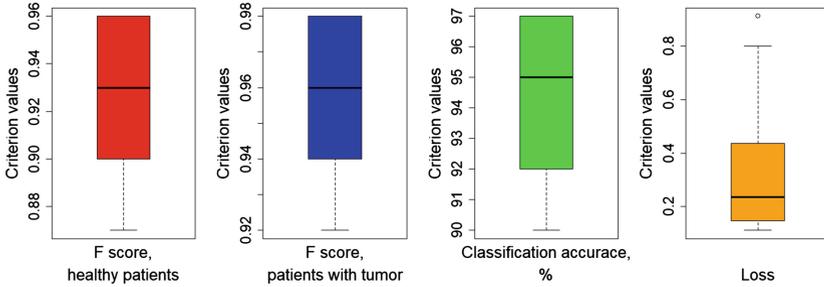


Fig. 4. Boxplots of the gene expression data classification quality criteria distribution when the 1D one level CNN were applied

The analysis of the obtained results allows us to conclude about the high efficiency of the application of this structure of CNN for the classification of objects based on gene expression data. The values of the F-index for both clusters, the objects classification accuracy are varied within a fairly high range, and the values of the loss function are relatively low in terms of the absolute values of gene expression. As it can be seen from Table 1, the best quality according to this group of criteria is achieved by applying the following parameters: filter size (length of the gene expression vector supplied to the network input) is 100, the full gene expression vector, in this case, is divided into 100 equal parts; density of the compacted layer is 256; kernel size is 8. When applying these parameters, the values of the quality criteria of the model are following: F-index for the first cluster - 0.96; F-index for the second cluster - 0.98; objects classification accuracy on the test subset is 97%, only one of the 39 objects was identified mistakenly; the value of the loss function is 0.112.

Figure 5 shows charts of both the objects classification accuracy and the values of the loss function change during the CNN training and model validation on each step of this procedure implementation. The analysis of the obtained results testifies to the correctness of the network training. Retraining is not observed, since the nature of the classification accuracy changes and the loss function calculated at the stages of the network training and model validation coincides within the range of the permissible error.

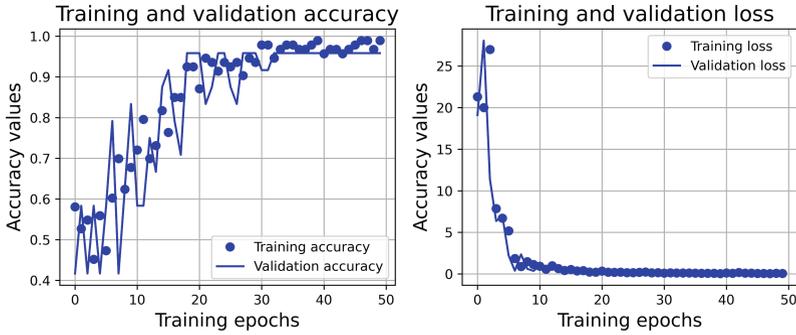


Fig. 5. Charts of the objects classification accuracy and the values of the loss function change during the CNN training and model validation

4.3 Application of 1D Two-Layer CNN for Gene Expression Data Classification

Figure 6 shows a block chart of a model of a one-dimensional two-layer convolutional neural network. Network hyperparameters, in this case, were set empirically. The simulation results have shown that increasing the kernel size by more than eight is not reasonable since the values of both the objects classification accuracy and the loss function values decrease when the kernel size increases. The length of the filter was (1×100) for the first layer and (1×50) for the second one. The density of the flatten layer varied within the range $(64, 128, 256, 512)$. The analysis of the simulation results allows concluding that when using a test subset of gene expression data in all cases 38 objects from 39 were correctly identified, the value of the F-index for the first and second clusters and the accuracy of object classification in all cases was the same and equal to 0.96, 0.98 and 0.97 respectively.

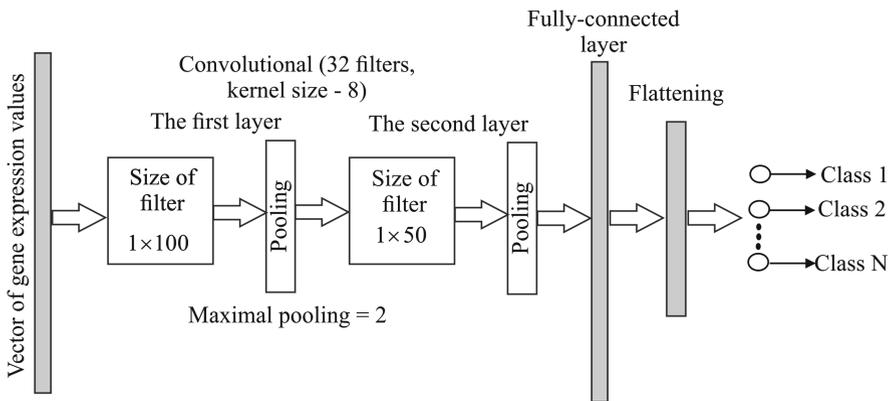


Fig. 6. Model of a one-dimensional two-layer convolutional neural network

Figure 7 shows the scatter plot of loss function values distribution when different values of flattening layer density were applied. An analysis of the obtained results allows us to conclude that the two-layer network is more stable in comparison with the one-layer one. Moreover, according to the loss function values, it is also more effective. The optimal value of the flattening layer where the minimum value of the loss function is reached 0.092, is equal to 256.

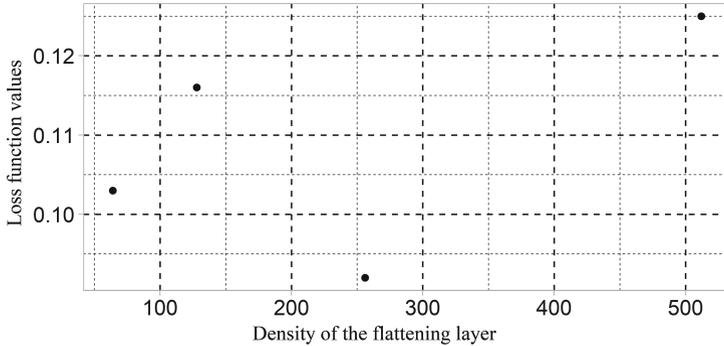


Fig. 7. A scatter plot of loss function values distribution when different values of flattening layer density were applied

Figure 8 shows charts of both the classification accuracy and the value of the loss function values which were calculated during the CNN training on data subsets used for network training and the model validation at each step of this procedure implementation. The analysis of the obtained diagrams also shows the adequacy of CNN training, since the nature of both the classification accuracy and loss function values changing on subsets of gene expression data used for training and the model validation coincides within the allowable range. Analysis of the results also has shown that increasing the number of layers is not reason-

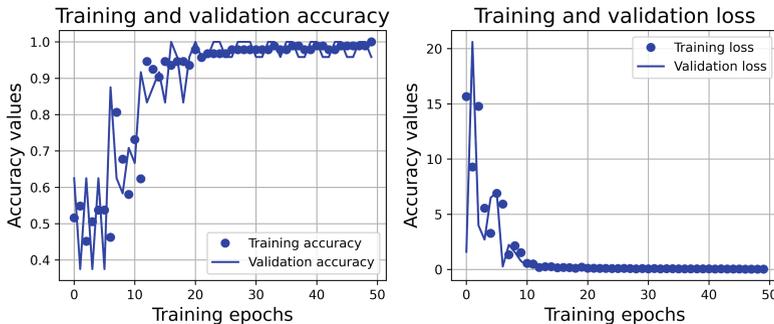


Fig. 8. Charts of the objects classification accuracy and the values of the loss function change during the 1D two-layer CNN training and model validation

able since the accuracy of object classification and the value of the loss function do not improve, but the learning time of the network increases.

4.4 Model of 2D Convolutional Neural Network

The model of a 2D convolutional network used to classify the objects based on the formed subset of gene expression data is shown in Fig. 9. In this case, the input data were formed as a matrix of gene expressions in size (100×100) . Filters (10×10) and (20×20) were applied to the data, and the maximum pooling of data in the convolutional layer was (2×2) . The simulation results have shown that reducing these values to (1×1) is not reasonable in terms of the applied quality criteria. The density of the flattening layer was increased within the range $(64, 128, 256, 512)$.

The simulation results are presented in Table 2. Analysis of the obtained results allows us to conclude that in terms of classification quality criteria, 2D CNN has a higher level of stability in comparison with the 1D network. In almost all cases (except one) the classification quality criteria are maximal ones, only one object was identified incorrectly. Moreover, the analysis of the results presented in Table 2 also allows us to conclude that reducing the size of the filter is not reasonable in terms of the loss function value. In all cases, the value of this function when applying the filter (10×10) is significantly higher in comparison with the use of filter (20×20) . According to the value of the loss function, the following network parameters are optimal in this case: density of the flattening layer - 64; the number of filters - 32; kernel size (20×20) .

Figure 10 shows the simulation results during both the CNN training and the model validation at each step of this procedure implementation. The analysis of these charts also indicates the correctness of the model. The consistency of the classification accuracy values and the loss function, calculated at the stages of both the model training and validation indicates the absence of the network retraining.

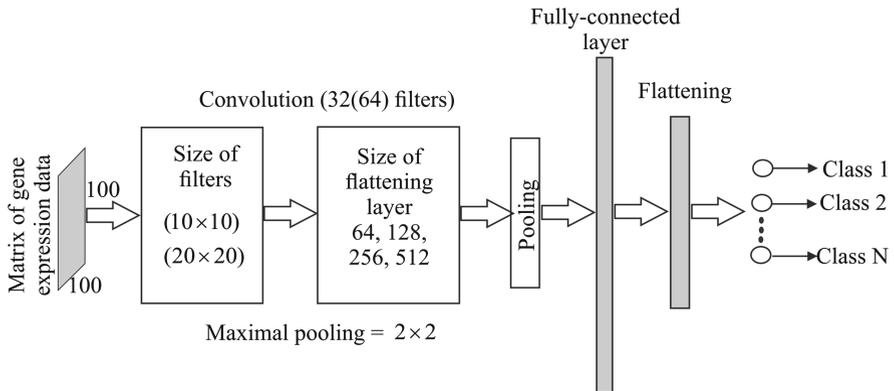


Fig. 9. Model of 2D convolutional neural network

Table 2. Simulation results regarding the use of 2D CNN

Hyperparameters			Classification results			
Flatten layer	Number of filters	Kernel size	F-index		Accuracy, %	Loss function values
			Cluster 1	Cluster 2		
64	32	(10,10)	0.96	0.98	97	1.24
64	64	(10,10)	0.96	0.98	97	1.738
64	32	(20,20)	0.96	0.98	97	0.688
64	64	(20,20)	0.96	0.98	97	1.08
128	32	(10,10)	0.96	0.98	97	1.94
128	64	(10,10)	0.96	0.98	97	2.363
128	32	(20,20)	0.96	0.98	97	1.161
128	64	(20,20)	0.96	0.98	97	1.412
256	32	(10,10)	0.96	0.98	97	3.032
256	64	(10,10)	0.93	0.96	95	4.079
256	32	(20,20)	0.96	0.98	97	1.318
256	64	(20,20)	0.96	0.98	97	1.470
512	32	(10,10)	0.96	0.98	97	3.335
512	64	(10,10)	0.93	0.96	95	4.898
512	32	(20,20)	0.96	0.98	97	1.225
512	64	(20,20)	0.96	0.98	97	1.604

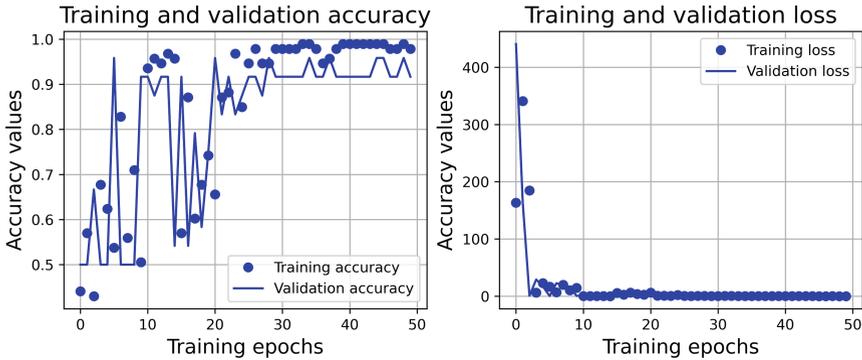


Fig. 10. Charts of the objects classification accuracy and the values of the loss function change during the 2D CNN training and the model validation

4.5 Model of 2D Three-Layer Convolutional Neural Network

Figure 11 shows the model of a two-dimensional three-layer convolutional neural network. The following values of the flattening layer density were studied during the simulation process: 64, 128, 256, 512. An analysis of the simulation results has shown that the value of classification accuracy on the test subset and F-index in all cases are the same: ACC = 97%, F-index for the first cluster was 0.96, and for the second one was 0.98. However, similar to other cases, 38 objects from 39 were identified correctly.

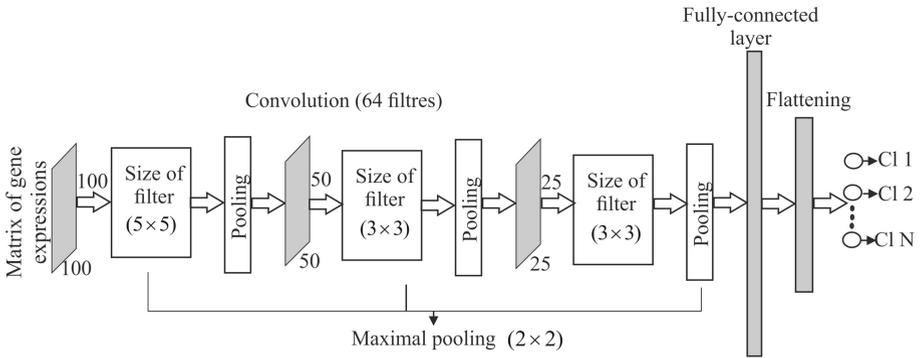


Fig. 11. Model of 2D three-layer convolutional neural network

Figure 12 shows the charts of the objects classification accuracy and loss function values calculated at the steps of both the model training and validation, the analysis of which also allows us to conclude that the network retraining is absent.

4.6 Estimation of CNN Robustness to Different Levels of Noise Component

To assess the CNN robustness to noise component, the Gaussian “white noise” with zero mean value was superimposed on the gene expression data, the standard deviation of which was varied within the range: 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4. This range was determined empirically by taking into account the values of the standard deviation of the gene expression profiles (Fig. 13), which were investigated during the simulation procedure implementation. The level of the noise component varied from the minimum (by

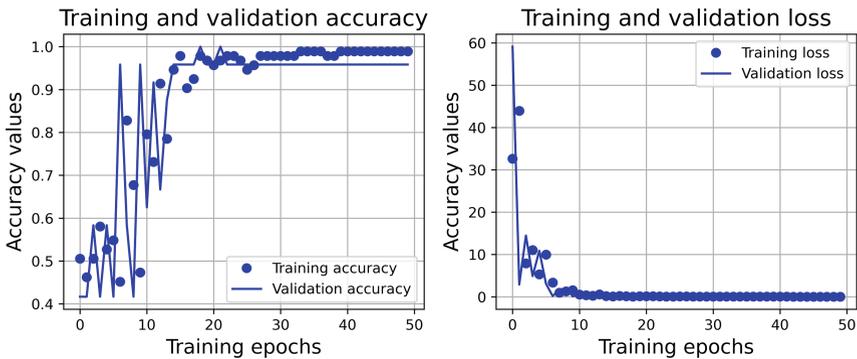


Fig. 12. Charts of the objects classification accuracy and the values of the loss function change during the 2D three-layer CNN training and the model validation

standard deviation) to the maximum, at which the noise component overlapped the useful component of the gene expression vector. When the noise component was superimposed on the gene expression data matrix, the negative values of the gene expressions (not correct) were replaced by zeros (this gene is not expressed for the corresponding object). We investigated 1D two-layer and 2D three-layer CNN with optimal values of hyperparameters which were determined hereinbefore. The robustness of the corresponding model was assessed by analysis of the objects classification accuracy calculated using the test subset of gene expression data.

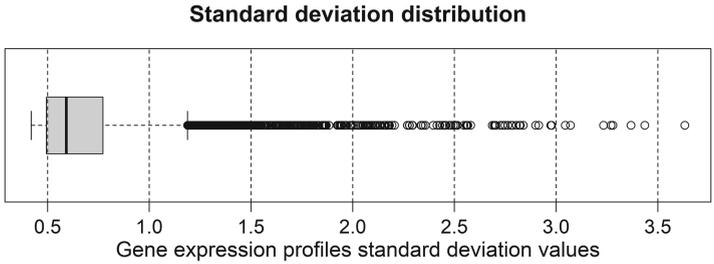


Fig. 13. Boxplot of the standard deviation values distribution of gene expressions of the studied profiles (10000 profiles)

Figure 14 shows the chart of the objects classification accuracy values versus the level of the noise component (standard deviation of the noise component).

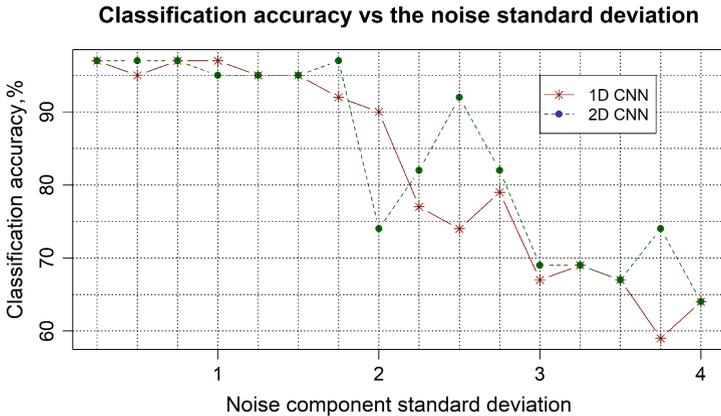


Fig. 14. Charts of the objects classification accuracy versus the level of the noise component when using one-dimensional and two-dimensional convolutional neural networks

An analysis of the obtained results allows us to conclude that both types of neural networks have a high level of robustness to low levels of noise component (for examined gene expression data the threshold level of network stability corresponded to the standard deviation of noise component 1.75). The classification accuracy of the test data subset varied within the range from 97% to 95%. One and two objects from 39 were identified incorrectly in the first and in the second cases respectively. This fact indicates that the classification models based on CNN are not sensitive to the length of the gene expression vector that characterizes the studied objects. Moreover, the simulation results allow us also to conclude that the one-dimensional CNN is more effective than the two-dimensional one because at almost the same robustness the one-dimensional network time operation is significantly less.

The obtained results create the conditions for the development of hybrid models for the mutually-correlated and differently-expressed gene expression data subset formation based on the joint application of clustering techniques and convolutional neural network. The evaluation of the quality of formed subsets of gene expression data is performed within the framework of the proposed model by calculating the appropriate objects classification quality criteria, and the formation of the desired length of gene expression values vector is performed by supplementing the data by vectors with zero values. The accuracy of the classification results, in this case, will not change significantly.

5 Conclusions

In the paper, we have presented the results of the research regarding the application of various structure convolutional neural networks for gene expression data classification. During the simulation process, we have investigated 1D and 2D one-layer and multi-layer networks with various combinations of CNN hyperparameters. The efficiency of the appropriate model was evaluated using classification quality criteria such as accuracy and F-score, which were calculated using the test data subset, and the loss function calculated at the stage of both the network training and the model validation. Gene expression dataset GSE19188 of patients studied for lung cancer has been used as the experimental ones during the simulation procedure implementation. These data were taken from the GSE (Gene Expression Omnibus) database and contain the results of DNA analysis of 156 patients using DNA microchip technology, where 65 patients were identified as healthy and 91 patients were diagnosed with cancer. Each microchip contained 54675 genes.

In the first stage, the number of genes was reduced from 54675 to 10000 by removing non-informative genes in terms of both statistical criteria and Shannon entropy. This vector of gene expression values (length 10000) was used as the input data at the stage of CNN implementation. Firstly, we have simulated 1D one-layer and two-layer CNN with various combinations of hyperparameters. The analysis of the obtained results allows us to conclude about the high efficiency of the application of this structure of CNN for the classification of objects

based on gene expression data. The values of the F-index for both clusters, the objects classification accuracy are varied within a fairly high range, and the values of the loss function are relatively low in terms of the absolute values of gene expression. Then, we have simulated 2D two-level and three-level CNN with various combinations of hyperparameters. Analysis of the obtained results has allowed us to conclude that in terms of classification quality criteria, 2D CNN has a higher level of stability in comparison with the 1D network. In almost all cases (except one) the classification quality criteria are maximal ones, only one object was identified incorrectly.

To evaluate the robustness to noise 1D and 2D CNN with optimal structure and parameters, we have generated noised vectors of gene expressions by stepwise increasing the noise component. The robustness of the corresponding model was assessed by analysis of the objects classification accuracy calculated using the test subset of gene expression data. An analysis of the obtained results has shown that both types of neural networks have a high level of robustness to low levels of the noise component. The classification accuracy of the test data subset varied within the range from 97% to 95%. One and two objects from 39 were identified incorrectly in the first and in the second cases respectively. This fact indicates that the classification models based on CNN are not sensitive to the length of the gene expression vector that characterizes the studied objects. Moreover, the simulation results have allowed us also to conclude that the one-dimensional CNN is more effective than the two-dimensional one because at almost the same robustness the one-dimensional network time operation is significantly less.

The further perspectives of the authors' research are the application of CNN within the hybrid models based on the joint use of clustering and classification techniques in order to allocate differently-expressed and mutually-correlated gene expression profiles.

References

1. Bioconductor. <https://www.bioconductor.org/>
2. Gene expression omnibus. <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi>
3. Babichev, S., Škvor, J.: Technique of gene expression profiles extraction based on the complex use of clustering and classification methods. *Diagnostics* **10**(8), 584 (2020). <https://doi.org/10.3390/diagnostics10080584>
4. Babichev, S., Lytvynenko, V., Škvor, J., et al.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction. In: *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018*, p. 8478452 (2018). <https://doi.org/10.1109/DSMP.2018.8478452>
5. Brownlee, J.: How to choose loss functions when training deep learning neural networks. *Deep learning performance* (2019). <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>
6. Brownlee, J.: Loss and loss functions for training deep learning neural networks. *Deep learning performance* (2019). <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/>

7. Busaleh, M., Hussain, M., Aboalsamh, H.: Breast mass classification using diverse contextual information and convolutional neural network. *Biosensors* **11**(11), 419 (2021). <https://doi.org/10.3390/bios11110419>
8. Cao, X., Pan, J.S., Wang, Z., et al.: Application of generated mask method based on mask R-CNN in classification and detection of melanoma. *Comput. Methods Programs Biomed.* **207**, 106174 (2021). <https://doi.org/10.1016/j.cmpb.2021.106174>
9. Chuang, Y.H., Huang, S.H., Hung, T.M., et al.: Convolutional neural network for human cancer types prediction by integrating protein interaction networks and omics data. *Sci. Rep.* **11**(1), 20691 (2021). <https://doi.org/10.1038/s41598-021-98814-y>
10. Hausser, J., Strimmer, K.: Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *J. Mach. Learn. Res.* **10**, 1469–1484 (2009)
11. Khan, S., Rahmani, H., Ali Shakh, S., Bennamoun, M.: *A Guide to Convolutional Neural Networks for Computer Vision*, p. 207. Morgan & Claypool Publishers, New England (2018)
12. Li, J., Sun, W., Feng, X., et al.: A dense connection encoding-decoding convolutional neural network structure for semantic segmentation of thymoma. *Neurocomputing* **451**, 1–11 (2021). <https://doi.org/10.1016/j.neucom.2021.04.023>
13. Lytvynenko, V., Savina, N., Krejci, J., Voronenko, M., Yakobchuk, M., Kryvoruchko, O.: Bayesian networks' development based on noisy-max nodes for modeling investment processes in transport. In: *CEUR Workshop Proceedings*, vol. 2386 (2019)
14. Marasanov, V.V., Sharko, A.V., Sharko, A.A.: Energy spectrum of acoustic emission signals in coupled continuous media. *J. Nano Electron. Phys.* **11**(3), 03028 (2019). [https://doi.org/10.21272/jnep.11\(3\).03028](https://doi.org/10.21272/jnep.11(3).03028)
15. Morandat, F., Hill, B., Osvald, L., Vitek, J.: Evaluating the design of the R language. In: Noble, J. (ed.) *ECOOP 2012. LNCS*, vol. 7313, pp. 104–131. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31057-7_6
16. Mostavi, M., Chiu, Y.C., Huang, Y., Chen, Y.: Convolutional neural network models for cancer type prediction based on gene expression. *BMC Med. Genom.* **13**(5), 44 (2020). <https://doi.org/10.1186/s12920-020-0677-2>
17. Philipsen, S., Hou, J., Aerts, J., et al.: Gene expression-based on classification of non-small cell lung carcinomas and survival prediction. *PLoS One* **5**(4), e10312 (2010)
18. Ramires, R., Chiu, Y., Horerra, A., et al.: Classification of cancer types using graph convolutional neural networks. *Front. Phys.* **8**, 203 (2020). <https://doi.org/10.3389/fphy.2020.00203>
19. Sharko, M., Shpak, N., Gonchar, O., Vorobyova, K., Lepokhina, O., Burenko, J.: Methodological basis of causal forecasting of the economic systems development management processes under the uncertainty. In: Babichev, S., Lytvynenko, V., Wójcik, W., Vyshemyrskaya, S. (eds.) *ISDMCI 2020. AISC*, vol. 1246, pp. 423–436. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-54215-3_27
20. Springenberg, J., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: the all convolutional net. In: *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings* (2015)

21. Yang, Y., Cai, J., Yang, H., Zhao, X.: Density clustering with divergence distance and automatic center selection. *Inf. Sci.* **596**, 414–438 (2022). <https://doi.org/10.1016/j.ins.2022.03.027>
22. Yu, Z., Yan, Y., Deng, F., Zhang, F., Li, Z.: An efficient density peak cluster algorithm for improving policy evaluation performance. *Sci. Rep.* **12**(1), 5000 (2022). <https://doi.org/10.1038/s41598-022-08637-8>



Formation of Subsets of Co-expressed Gene Expression Profiles Based on Joint Use of Fuzzy Inference System, Statistical Criteria and Shannon Entropy

Igor Liakh¹ , Sergii Babichev^{2,3} , Bohdan Durnyak⁴ , and Iryna Gado⁵ 

¹ Uzhhorod National University, Uzhhorod, Ukraine
ihor.lyah@uzhnu.edu.ua

² Jan Evangelista Purkyně University in Ústí nad Labem,
Ústí nad Labem, Czech Republic
sergii.babichev@ujep.cz

³ Kherson State University, Kherson, Ukraine
sbabichev@ksu.ks.ua

⁴ Ukrainian Academy of Printing, Lviv, Ukraine
durnyak@uad.lviv.ua

⁵ Lviv Polytechnic National University, Lviv, Ukraine
iryna.v.nychai@lpnu.ua

Abstract. The paper presents the results of the research regarding the application of a fuzzy logic inference system to form the co-expressed gene expression profiles based on the joint use of Shannon entropy and statistical criteria. The allocation of co-expressed genes can allow us to increase the disease diagnosis accuracy on the one hand and, reconstruct the qualitative gene regulatory networks on the other hand. To solve this problem, we have proposed the joint use of the fuzzy logic inference system and random forest classifier. In beginning, we have calculated for each of the gene expression profiles the maximum expression values, variance and Shannon entropy. These parameters were used as the input ones for the fuzzy logic inference system. After setting the fuzzy membership functions for both the input and output parameters, the model formalization including fuzzy rules formation, we have applied the model to gene expression data which included initially the 54675 genes for 156 patients examined at the early stage of lung cancer. As a result of this step implementation, we have obtained the four subsets of gene expression profiles with various significance levels. To confirm the obtained results, we have applied the classification procedure to investigated samples that included as the attributes the allocated genes. The analysis of the classification quality criteria allows us to conclude about the high effectiveness of the proposed technique to solve this type of task.

Keywords: Gene expression profiles · Fuzzy model · Statistical criteria · Shannon entropy · Co-expressed genes · Classification quality criteria · ROC analysis

1 Introduction

Currently, the method of sequencing RNA molecules is the most effective for generating gene expression data. The use of this method allows us to determine with sufficient accuracy the number of genes that correspond to the investigated object. It is obvious that the number of genes is proportional to the value of the gene expression and, this value determines the level of this gene activity. The greater gene expression value corresponds to the greater amount of this gene identified for the appropriate biological entity. Typically, approximately 25000 genes are active in the human genome. Reconstructing the gene regulatory network based on a complete set of genes is problematic since: first, it requires large resources; secondly, the interpretation of such a network is very difficult. Taking into account the analysis of current gene expression databases, a large number of genes are low-expressed for all studied objects and they can be deleted without significant loss of information. The implementation of this step is possible by applying various quantitative statistical and entropy criteria which allow dividing genes into informative and non-informative ones. Usually, the number of informative genes is reduced by about half and will be equal to $\approx 7000 \div 10000$. However, the reconstruction of the gene regulatory network based on such a number of genes is also problematic. Therefore, there is a necessity to form subsets of co-expressed genes based on a joint application of both current data mining and classification techniques of objects that contain gene expression values in selected clusters as attributes.

2 Problem Statement

Gene expression data used for gene regulatory networks reconstruction are usually presented as a matrix $e_{i,j}$, $i = \overline{1,n}$, $j = \overline{1,m}$, where n and m are the numbers of genes and studied objects respectively. After deleting zero-expressed genes for all objects (unexpressed genes), approximately 25000 genes remain that define the genome of the corresponding biological organism. It should be noted that a large number of genes are lowly expressed for all objects, they determine certain processes occurring in the biological organism, but are not decisive in terms of the health status of the object (disease identified and studied). Therefore, in the first step, it is advisable to remove low-expressed genes for all objects. In the second step, it is advisable to remove genes whose expression values changed slightly when analyzing different types of objects (by variance or standard deviation) or change randomly (high Shannon entropy value), which corresponds to noise. These gene expression profiles do not allow us by the level of expression to unambiguously identify relevant objects according to the health status of the biological organism, and they can also be deleted from the database. The gene expression profile in this case means the vector of expression values of the corresponding gene, which are determined for all investigated objects. In the context of this definition, co-expressed genes are genes whose expression values change accordingly for all studied objects. At the same time, the profiles of

co-expressed genes allow identifying objects with high accuracy considering the state of health of the biological organism.

The procedure for generating co-expressed gene expression profiles assumes two stages. The first stage is to reduce the number of genes with low absolute value in the first step and low variance and high Shannon entropy in the second step. This raises the problem of determining the threshold coefficient for each of the used criteria. Typically, the threshold coefficients are determined empirically during the simulation process implementation taking into account the approximate number of genes that should remain after the implementation of this stage. However, taking into account the fact that co-expressed gene expression profiles should allow the identification of objects with the highest possible accuracy, the values of the threshold coefficients for each of the criteria can be determined by the maximum value of the classification accuracy of the studied objects. Thus, the implementation of the concept of the formation of co-expressed gene expression profiles assumes the use of a hybrid model that involves the joint use of both data mining methods and machine learning techniques.

3 Literature Review

A lot of scientific works are devoted to forming subsets of co-expressed genes based on various types of quantitative and/or qualitative criteria nowadays. So, in [3], the authors proposed a technique of big data feature reduction based on the forward selection depending upon map-reduce. The proposed method was based on the application of a linear-based Support Vector Machine (SVM) using map-reduce based classifier to order the microarray information. The performed calculations were executed on the Hadoop system, and a relative investigation was finished using various datasets. The main disadvantages of the presented research are absent comparison analysis of similar results using other classification techniques and other kernel functions when applying the SVM method.

The paper [10] focused on the research in the field of dimensionality reduction to overcome the dimensionality disaster of omics data. The authors have proposed a random forest deep feature selection (RDFS) algorithm. The proposed hybrid model is based on the joint use of a random forest classifier and deep neural network. The simulation results have shown that the objects classification results were improved due to reducing the dimensions of multi-omics data by integrating gene expression data and copy number variation data. The results of the simulation have shown also that the classification accuracy value and area under the ROC curve when the multi-omics data were used are better than in the case of the use of single-omics data under the RDFS algorithm application.

The research regarding the creation of a novel feature selection algorithm on the basis of both the feature individual distinguishing ability and feature influence in the biological network has been proposed in [15] in order to determine the important biomolecules to identify different disease conditions. In this research, the feature distinguishing ability has been evaluated using the overlapping area of the feature effective ranges in various classes. The features' comprehensive

weight was obtained by combining the features' individual distinguishing abilities and features' influence on the network. The authors have shown that evaluating biomolecules from the molecular level and network level is useful and has high performance for identifying the potential disease biomarkers.

However, we would like to note that despite achievements in this subject area, the problem of feature selection when the dimension of big data is reduced has no obvious solution nowadays. To our mind, this problem can be solved by the development of hybrid models containing modern both data mining and machine learning techniques, which are used, which are used successfully in various areas of scientific research nowadays [12, 13, 16].

The goal of the research is the development of a hybrid model of co-expressed gene expression profiles extraction based on the joint use of fuzzy logic inference system and classification technique.

4 Fuzzy Model of Removing the Non-informative Gene Expression Profiles by Statistical and Entropy Criteria

The problem of removing non-informative gene expression profiles by statistical and entropy criteria was solved in [4, 5]. In the authors' mind, the gene expression profile was considered informative if the maximum value of the expression of this profile and variance is greater, and Shannon's entropy is less than the corresponding threshold values:

$$\{e_{ij}\} = \left\{ \begin{array}{l} \max_{i=1, n}(e_{ij} \geq e_{bound}), \text{ and } var(e_{ij}) \geq var_{bound}, \\ \text{and } entr(e_{ij}) \leq entr_{bound} \end{array} \right\}, j = \overline{1, m} \quad (1)$$

where: n is the number of samples or objects to be examined; m is the number of genes.

The boundary values, in this case, were also determined empirically during the simulation process, taking into account the approximate number of genes that should make up a subset of experimental data for further simulation. However, it should be noted that the concept proposed by the authors has a significant drawback. A high value of the variance of the corresponding gene expression profile or a low Shannon entropy value (according to these criteria, this gene expression profile is considered informative) when low absolute values of gene expression for all studied objects does not mean that this gene expression profile is informative since, in terms of absolute values, this gene does not contribute to the high accuracy of identification of the studied objects. Thus, there is a necessity to set priorities for the relevant operations, either by entering the sequence of their application or by initializing the weights of a particular operation. However, in this instance, it is necessary to justify the choice of the value of the appropriate weight.

Within the framework of our research, this problem is solved on the basis of fuzzy logic inference system application [8, 17–19], and the priority of one or another operation is taken into account when creating a base of fuzzy rules that

form the basis of fuzzy model. The formation of fuzzy rules requires the following steps:

- define the set of input variables: $X = x_1, x_2, \dots, x_n$ with the corresponding terms for each variable: $T_{input} = t_i^p$, $i = \overline{1, n}$, $p = \overline{1, q}$, where q is the number of terms corresponding to the i -th input variable;
- define the set of output variables: $T_{out} = t^r$, $r = \overline{1, q}$, where q , in this case, is the number of terms corresponding to the output variable;
- generate a finite set of fuzzy rules agreed with appropriate input and output variables:

$$\bigcup_{k=1}^m \left[\bigcap_{p=1}^q (x_i = t_p^q), \text{ when } \omega_k \right] \longrightarrow (y = t^r), \quad i = \overline{1, n}, \quad r = \overline{1, q} \quad (2)$$

where: $k = \overline{1, m}$ is a number of fuzzy rules that make up a fuzzy database; ω_k is the weight of the k -th rule (determined in the case of priority rules existence).

In the general case, the fuzzy inference procedure involves the following steps:

- *fuzzification* or matching between the specific values of the input variables used in the model and the values of the corresponding membership functions, taking into account the relevant term corresponding to this membership function. At the stage of fuzzification, the membership functions, which are predetermined on the input variables, are applied to their input values, in other words, the values of the membership functions $\mu_i^{t_i^k}(x_i)$ of the input variable x_i for the term t_i^k are determined. The result of the fuzzification step implementation is a matrix of values of membership functions for all input variables, which are defined for all fuzzy rules included in the fuzzy database;
- *aggregation* or determination of the degree of truth of the conditions for each of the fuzzy rules by finding the level of “clipping” for the preconditions of each rule using the operation *min*:

$$\alpha_k = \bigwedge_{i=1}^n [\mu_i^{t_i^k}(x_i)] \quad (3)$$

- *activating* or finding the degree of truth for each of the fuzzy rules by forming the truncated membership functions of the fuzzy sets for each of the fuzzy rules:

$$\mu'_k(y) = \alpha_k \wedge \mu_k(X) \quad (4)$$

where: $\mu_k(X)$ are the truncated membership functions for the vector of input variables corresponding to the k -th rule; $\mu'_k(y)$ is the resulting membership function for the output variable, which corresponds to the k -th rule;

- *accumulation* or formation of the membership function of the resulting fuzzy set for the output variable using the operation *max*:

$$\mu^{\Sigma}(y) = \bigvee_{k=1}^m [\mu'_k(y)] \quad (5)$$

- *defuzzification* or finding a crisp value of the output variable by applying the appropriate operation to the obtained membership function of the resulting fuzzy set. The defuzzification operation can be implemented using various methods: calculation of the obtained function gravity centre, the centre of the figure area, and the left or right modal values. Within the framework of the proposed model, the most common centre of gravity method has been used:

$$Y = \frac{\int_{min}^{max} y \cdot \mu_{\Sigma}(y) dy}{\int_{min}^{max} \mu_{\Sigma}(y) dy} \quad (6)$$

The practical implementation of the fuzzy inference model within the framework of the research assumes the following steps:

1. Determining the ranges of variation of the values of the input statistical criteria, Shannon entropy and the output parameter (the significance of the profile in the ability to identify the object).
2. Defining the fuzzy sets membership functions for input and output parameters.
3. Formation of a base of fuzzy rules that form a fuzzy inference.
4. Choice of fuzzy inference algorithm and method to form the crisp value of output variable.
5. Determining the quantitative criteria for assessing the adequacy of the model for its testing.

The range of the input parameters values variation within the proposed model was determined by analyzing the general statistics, while the absolute values of gene expressions in the first step were determined by the maximum value of expression for each profile. Then, the general statistic was formed for the obtained vector of maximum values of gene expressions, vector of variance of gene expression profiles and Shannon entropy. To create a fuzzy model, the inter-quantile ranges of the appropriate criteria values variation were used. The formed ranges were divided into three subranges with the corresponding terms. For variance and maximum absolute values of gene expressions, these ranges were the following: $0\% \leq x < 25\%$ – “Low” (Low); $25\% \leq x < 75\%$ – “Medium” (Md); $x \geq 75\%$ – “High” (Hg). For Shannon’s entropy: $x \geq 75\%$ – “High” (Hg); $25\% \leq x < 75\%$ – “Medium” (Md); $x < 25\%$ – “Low” (Low). The range of the output parameter variation (significance of the profile) in the proposed models varied from 0 to 100 and was divided into five equal intervals: $0\% \leq x < 20\%$ – “Very low” (VLow); $20\% \leq x < 40\%$ – “Low” (Low); $40\% \leq x < 60\%$ – “Medium” (Md); $60\% \leq x < 80\%$ – “High” (Hg); $80\% \leq x \leq 100\%$ – “Very high” (VHg). Regarding the fuzzy sets membership functions, the trapezoidal membership functions were used for the input parameters for the values with the terms “Low” and “High”, and the triangular membership functions was used for the medium range of values (Md). The triangular fuzzy sets membership functions were applied for all subranges of the output parameter. It should be noted that the parameters of the fuzzy sets membership functions of input parameters involve an adjustment during the simulation process implementation taking into

account the nature of the distribution of gene expression values in the studied experimental data.

Table 1 presents the terms of the fuzzy rules base which were used during the fuzzy model creation.

Table 1. Terms of the fuzzy model knowledge base to form the subsets of co-expressed gene expression profiles

No	Maximum expr. value	Variance	Shannon entropy	Significance of profile
1	Hg	Hg	Low	VHg
2	Hg	Md	Low	Hg
3	Hg	Hg	Md	Hg
4	Hg	Md	Md	Hg
5	Hg	Low	Md	Hg
6	Hg	Hg	Hg	Hg
7	Hg	Low	Hg	Md
8	Hg	Md	Hg	Md
9	Md	Hg	Low	Hg
10	Hg	Low	Low	Hg
11	Md	Md	Low	Md
12	Md	Low	Low	Md
13	Md	Hg	Md	Md
14	Md	Md	Md	Md
15	Md	Low	Md	Md
16	Md	Hg	Hg	Md
17	Md	Low	Hg	Low
18	Low	Hg	Low	Low
19	Low	Md	Low	Low
20	Low	Hg	Md	Low
21	Low	Low	Low	Low
22	Low	Md	Md	Low
23	Low	Low	Hg	VLow

As can be seen from Table 1, the priority parameter for identifying the significance of the gene expression profile is the maximum of the gene profile expression values, which are determined for all studied objects. As noted hereinbefore, genes whose expression values are relative low for all objects are not decisive to the identification of objects and can be deleted despite high variance and/or low Shannon entropy values. The combination of Shannon entropy and variance values, in this case, are corrective ones.

Fuzzy logical equations linking the values of the membership functions of the input and output variables in the proposed model can be represented as follows:

$$\mu^{VHg}(y) = \mu^{Hg}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Low}(entr) \quad (7)$$

$$\begin{aligned} \mu^{Hg}(y) = & [\mu^{Hg}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Hg}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Hg}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Hg}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Hg}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Md}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Hg}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Hg}(entr)] \end{aligned} \quad (8)$$

$$\begin{aligned} \mu^{Md}(y) = & [\mu^{Hg}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Hg}(entr)] \vee \\ & [\mu^{Md}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Md}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Md}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Md}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Md}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Hg}(entr)] \vee \\ & [\mu^{Hg}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Hg}(entr)] \vee \\ & [\mu^{Md}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Low}(entr)] \end{aligned} \quad (9)$$

$$\begin{aligned} \mu^{Low}(y) = & [\mu^{Md}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Hg}(entr)] \vee \\ & [\mu^{Low}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Low}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Low}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Low}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Low}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Md}(entr)] \end{aligned} \quad (10)$$

$$\mu^{VLow}(y) = \mu^{Low}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Hg}(entr) \quad (11)$$

The membership function of the final fuzzy subset for the output variable “Significance of the profile” is formed according to the following equation:

$$\mu^{\Sigma}(y) = \mu^{VHg}(y) \vee \mu^{Hg}(y) \vee \mu^{Md}(y) \vee \mu^{Low}(y) \vee \mu^{VLow}(y) \quad (12)$$

The last step determines the crisp value of the output variable as the gravity center of the resulting figure in accordance with formula (6).

4.1 Simulation Regarding Practical Implementation of the Proposed Fuzzy Logic Inference Model

Approbation of the proposed technique of forming groups of co-expressed gene expression profiles based on both the statistical criteria and Shannon entropy was carried out using the gene expressions dataset of patients studied in the early stages of lung cancer. GSE19188 data [9] were taken from the freely available Gene Expression Omnibus database [2] and contained gene expression data from 156 patients, of whom 65 were identified as healthy in clinical trials and 91 had early-stage cancers. Data processing was performed by using the tools of the Bioconductor package [1] of the programming language R [14]. In the initial state, the data contained 54675 genes. In the first stage, for each gene expression profile, the maximum value of gene expressions, variance and Shannon entropy was calculated using the James-Stein shrinkage estimator. Figure 1 shows the box plots of the obtained values distribution, the analysis of which allows us to conclude regarding the reasonable of using the hereinbefore set ranges of relevant criteria values for setting up a fuzzy logic inference model. Really, the most informative gene expression profiles have high values of expression and variance and low values of Shannon entropy.

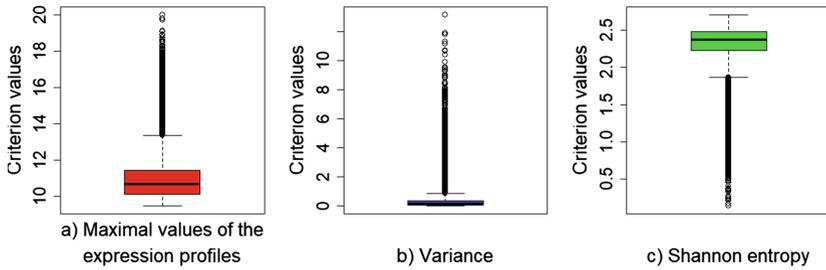


Fig. 1. The nature of the distribution of statistical criteria and Shannon entropy of gene expression profiles of patients studied for early-stage lung cancer

Figure 2 shows the membership functions of both the input and the output parameters fuzzy sets used in the proposed model. Figure 3 presents the step-wise procedure described hereinbefore and implemented during the simulation procedure that allows us both to form the subsets of co-expressed gene expression profiles and to evaluate the effectiveness of the proposed fuzzy model by analyzing the results of investigated objects classification that contain, as the attributes, formed subsets of gene expression data.

As it can be seen in Fig. 3, the implementation and evaluation of the fuzzy logic inference model involve the following stages:

Stage I. Formation of quality criteria vectors of gene expression profiles and general statistics of the obtained vectors.

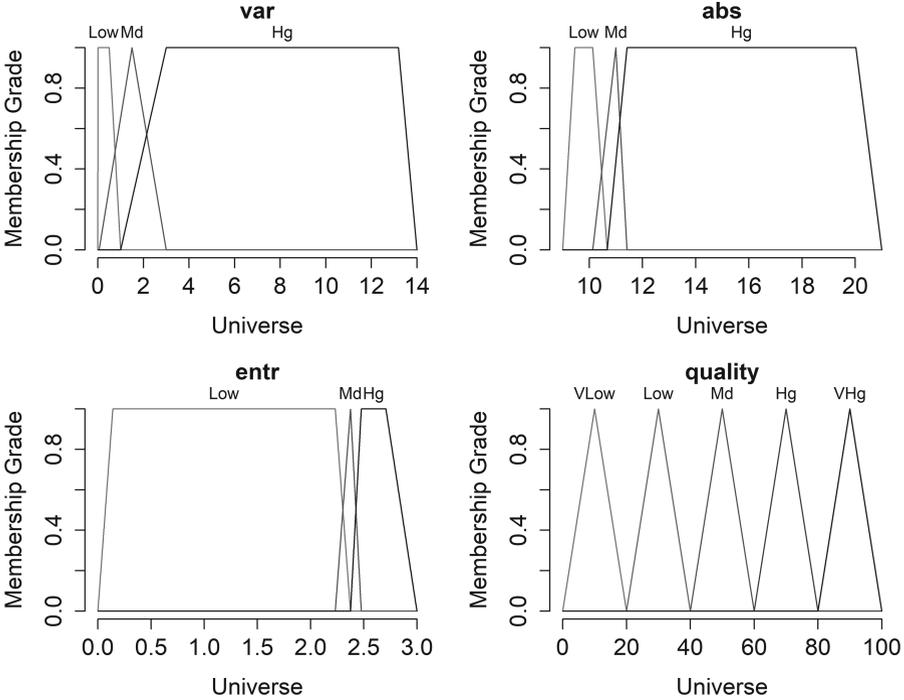


Fig. 2. The membership functions of fuzzy sets of input and output parameters used in the fuzzy model of generating co-expressed gene expression profiles

- 1.1. The calculation for each gene expression profile of the expression maximum value, variance and Shannon entropy. The length of the obtained vectors is equal to the number of genes that are contained in the experimental data.
- 1.2. Calculation of general statistics of the obtained vectors. Fixation of the ranges of the appropriate values variation and quantiles corresponding to 25, 50 and 75% of variation of the appropriate range.

Stage II. Creation, debugging and implementation of fuzzy logic inference system.

- 2.1. Formation of the fuzzy logic inference system structure formalization of the model, determination of the fuzzy sets membership functions for input and output parameters, formation of the model fuzzy rules base.
- 2.2. Application of fuzzy logic inference model to gene expression profiles, formation of subsets of co-expressed gene expression profiles according to the level of their significance taking into account the corresponding values of statistical criteria and Shannon entropy.

Stage III. Assessing the fuzzy model adequacy by applying a classifier to objects that contain, as attributes, subsets of allocated gene expression data.

- 3.1. The choice of classifier considering the type of the experimental data, and the formation of classification quality criteria.

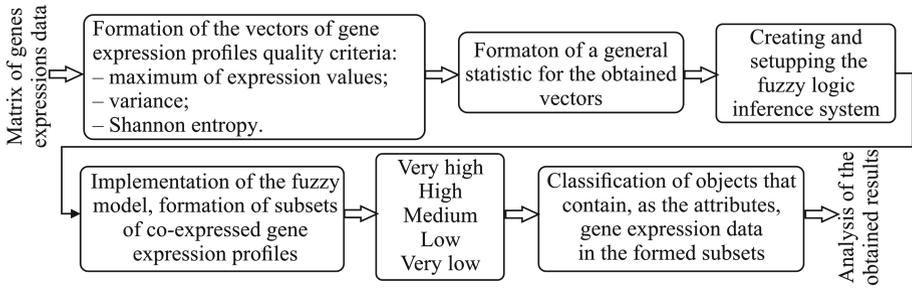


Fig. 3. Structural block chart of a stepwise procedure to form subsets of co-expressed gene expression profiles based on the joint use of fuzzy logic inference system and objects classification technique

3.2. Implementation of the objects classification procedure. The objects, in this case, contain, as attributes, gene expression data in the formed subsets.

3.3. Calculation of the classification quality criteria.

Stage IV. Analysis of the obtained results.

4.1. Making appropriate decisions concerning the adequacy of the fuzzy logic inference model operation, taking into account the correlation between the results of the objects classification based on gene expression data in the formed subsets and the level of the respective gene expression profiles significance.

Figure 4 presents the results of the simulation regarding the application of the fuzzy inference model for the formation of gene expression profiles subsets of the different significance levels. Considering that only 29 genes from 54675 ones were identified as “Very High” significance, the groups containing genes with “Very High” and “High” significance levels were pooled for further simulation. The analysis of the obtained results allows us to conclude about the adequacy of the proposed fuzzy inference model for dividing a set of genes into corresponding subsets by the number of genes. It is well known that the human genome consists of approximately 25000 active genes. From this point of view, the allocation of 13734 genes of very high and high significance and 13096 genes of medium significance for further processing is reasonable. Genes with low and very low significance can be removed from the data as uninformative ones.

The next stage that may confirm or refute the conclusion regarding the adequacy of the results obtained by applying the fuzzy inference system to remove the non-informative gene expression profiles according to the used criteria is to apply a classifier to identify objects that contain, as attributes, the allocated gene expression data in the corresponding subsets.

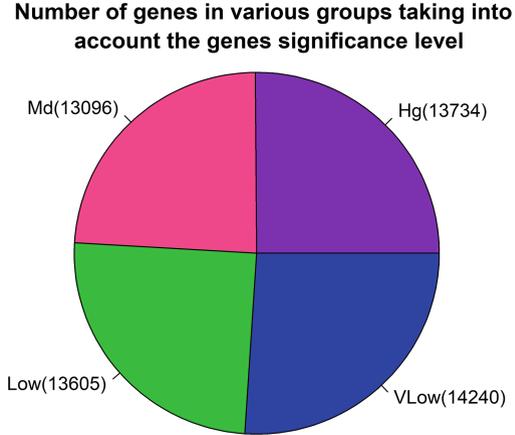


Fig. 4. Simulation results regarding the application of the fuzzy logic inference model for the formation of subsets of gene expression profiles of different significance levels according to statistical criteria and Shannon entropy

5 Assessing the Fuzzy Inference Model Adequacy by Applying the Gene Expression Data Classification Technique

The assessment of the objects classification quality within the framework of the research was carried out using errors of both the first and second kind. The following criteria were applied as the quality criteria:

- objects classification accuracy:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

where: TP and TN are the correctly identified positive and negative cases respectively (for example, the presence or absence of the disease); FP and FN are the mistakenly identified positive and negative cases (the errors of the first and the second kind);

- F-measure is defined as the harmonic average of Precision (PR) and Recall (RC):

$$F = \frac{2 \cdot PR \cdot RC}{PR + RC} \tag{14}$$

where:

$$PR = \frac{TP}{TP + FP}; \quad RC = \frac{TP}{TP + FN}$$

- Matthews Correlation Coefficient (MCC):

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{15}$$

Table 2 presents the procedure for forming a confusion matrix, which is the basis for calculating the classification quality criteria in accordance with formulas (13)–(15).

Table 2. The confusion table to identify the errors of the first and the second kind

State of the object by the results of clinical testing	Results of the objects classification	
	Patient(True - 1)	Healthy(False - 0)
Patient(True - 1)	TP(True Positives)	FN(False Negatives)
Healthy(False - 0)	FP(False Positives)	TN(True Negatives)

In this instance, the classification accuracy is maximal one (100%) if all objects are correctly identified and errors of the first (FP) and second (FN) kind are absent. The values of the F-index and MCC criterion are also maximal ones and they are equal to 1.

The second type of criterion which was used in the research to assess the object classification quality is based on ROC (Received Operating Characteristic) analysis. Application of this criterion assumes the calculating the area under the ROC curve AUC (Area Under Curve). A larger area corresponds to a higher quality of the object classification.

The choice of the classifier is determined by the peculiarity of experimental data. When the gene expression data is used as the experimental ones, the key feature is a large number of attributes (the number of genes that determine the state of the studied object). As was noted hereinbefore, 156 patients were used as gene expression data during the simulation process, of which 65 were identified as healthy by the clinical trials and 91 were identified as ill with early-stage of cancers tumour. The initial number of genes (54675) was divided into four groups according to statistical criteria and Shannon entropy (Fig. 4). Each group contained approximately 13000 genes which were used as the input data of the classifier. In this case, the classifier should be focused on big data. In [6, 7, 11], the authors present the results of research regarding the use of convolutional neural networks as a classifier for the identification of objects based on gene expression data. The authors investigated different topological structures of this type of network and proved their effectiveness for the classification of objects based on high-dimensional gene expression data. However, it should be noted that the correct use of convolutional neural networks involves the formation of convolutional layers to supplement the data with profiles with zero expression values to obtain the required number of genes. In the current research, this step may affect the results, which is undesirable. For this reason, the use of convolutional neural networks at this stage of simulation is not reasonable. In [4, 5], the authors presented the results of studies focused on comparing binary classifiers to identify objects based on high-dimensional gene expression data. The authors have shown that the Random Forest Binary Classification Algorithm is more

efficient for identifying objects based on gene expression data than other similar classifiers. For this reason, this classifier was used in the current research.

The simulation results regarding the identification of objects that contain gene expression data as attributes are presented in Table 3 and Fig. 5. An analysis of the obtained results confirms the expediency of using the proposed fuzzy logic inference model for the formation of subsets of gene expression profiles of different significance levels according to statistical criteria and Shannon entropy. The values of the classification quality criteria presented in Table 3 gradually increase with the transition from subsets of gene expression profiles with a very low significance level to a subset of gene expression profiles with a high significance level. The F-measure and MCC criterion values, in these cases, also match within the margin of admissible error.

Table 3. The results of the simulation regarding the classification of objects based on gene expression data of various significance level

Significance of genes	Classification quality criteria				
	Accuracy,%	Sensitivity	Specifity	F-measure	MCC
High	98.4	1	0.973	0.992	0.967
Medium	93.5	1	0.9	0.967	0.873
Low	90.3	0.917	0.917	0.894	0.801
Very Low	85.5	0.870	0.846	0.862	0.701

An analysis of the AUC criterion values (Area Under ROC-curve) also confirms the feasibility of using a fuzzy inference system to divide gene expression profiles into subsets of genes of various significance levels. However, it should be noted that this type of criterion is significantly less sensitive for gene expression profiles, which are allocated into the subsets with high and medium significance levels. Moreover, the AUC criterion value for a subset of genes with very low significance is higher than the similar value for the subset of gene expression profiles with low significance, which is not correct and contradicts the values of the classification quality criteria presented in Table 3. However, it should be noted that this criterion allows us to divide the set of gene expression profiles into two subsets: a subset of informative genes in this case contain genes with high and medium significance levels; other genes are removed as uninformative ones.

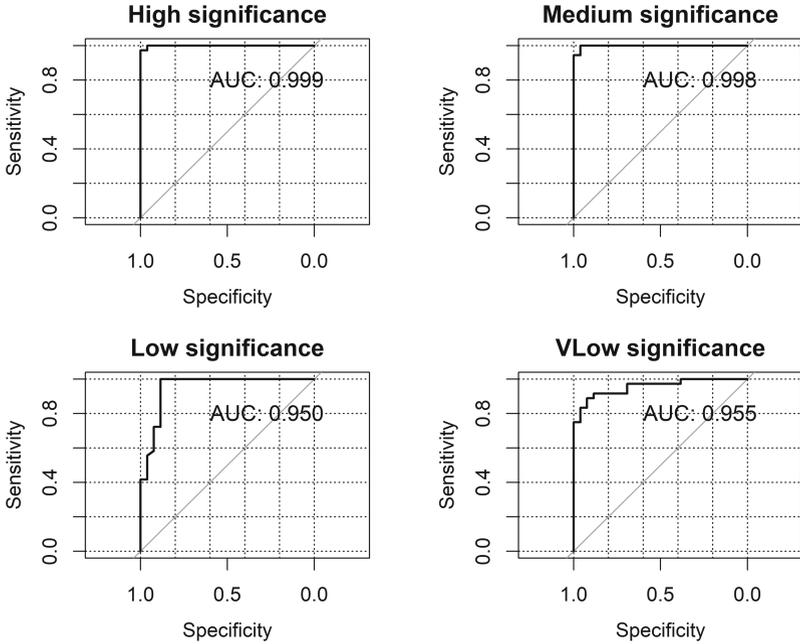


Fig. 5. The results of ROC analysis to assess the effectiveness of a fuzzy model for the formation of subsets of gene expression profiles by the level of their significance

6 Conclusions

In this manuscript, we have presented the results of the research regarding the creation of a hybrid model of co-expressed gene expression profiles formation based on the complex use of fuzzy logic inference system and classification technique. The proposed model is presented as a step-wise procedure of gene expression data processing. The statistical criteria (maximum expression value and variance) and Shannon entropy were used as the input parameters of the fuzzy logic inference model and, the gene expression profiles significance level was used as the output parameter. To create a fuzzy model, the inter-quantile ranges of the appropriate criteria values variation were used. The formed ranges were divided into three subranges with the corresponding terms: low, medium, and high. The range of the output parameter variation (significance of the profile) in the proposed models varied from 0 to 100 and was divided into five equal intervals. The trapezoidal membership function was used for the input parameters for the values with the terms “Low” and “High”, and the triangular membership functions were used for the medium range of values (Md). The triangular fuzzy sets membership functions were applied for all subranges of the output parameter.

The knowledge database has been presented in the form of a system of fuzzy equations, which have been used during the fuzzy inference procedure implementation. The gene expressions dataset of patients studied in the early stages

of lung cancer was used as the experimental one. This data set contained gene expression data from 156 patients, of whom 65 were identified as healthy in clinical trials and 91 had early-stage cancers. In the initial state, the data contained 54675 genes. The analysis of the obtained results regarding the fuzzy logic model implementation allows us to conclude its adequacy for dividing a set of genes into corresponding subsets by the number of genes since the allocation of 13734 genes of very high and high significance and 13096 genes of medium significance for further processing is reasonable in terms of a number of genes in humans genome (approximately 25000).

To confirm the proposed fuzzy model adequacy, we have applied the objects classification technique with the calculation of various types of classification quality criteria. A random forest classifier has been used during the simulation process at this stage. An analysis of the obtained results confirms the expediency of using the proposed fuzzy logic inference model for the formation of subsets of gene expression profiles of different significance levels according to statistical criteria and Shannon entropy. The values of the classification quality criteria have gradually increased with the transition from subsets of gene expression profiles with a very low significance level to a subset of gene expression profiles with a high significance level. An analysis of the Area Under ROC-curve (AUC) values also confirms the feasibility of using a fuzzy inference system to divide gene expression profiles into subsets of genes of various significance levels. However, it should be noted that this type of criterion is significantly less sensitive for gene expression profiles, which are allocated into the subsets with high and medium significance levels. Moreover, the AUC criterion value for a subset of genes with very low significance is higher than the similar value for the subset of gene expression profiles with low significance, which is not correct and contradicts the values of other used classification quality criteria. However, it should be noted that this criterion allows us to divide the set of gene expression profiles into two subsets: a subset of informative genes in this case contains genes with high and medium significance levels; other genes are removed as uninformative ones.

References

1. Bioconductor. <https://www.bioconductor.org/>
2. Gene expression omnibus. <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi>
3. Abinash, M., Vasudevan, V.: Gene data classification using map reduce based linear SVM. *Concurrency Comput. Pract. Experience* **34**(8), e5497 (2022). <https://doi.org/10.1002/cpe.5497>
4. Babichev, S., Škvor, J.: Technique of gene expression profiles extraction based on the complex use of clustering and classification methods. *Diagnostics* **10**(8), 584 (2020). <https://doi.org/10.3390/diagnostics10080584>
5. Babichev, S., Lytvynenko, V., Škvor, J., et al.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction. In: *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018*, pp. 36-341 (2018). <https://doi.org/10.1109/DSMP.2018.8478452>

6. Busaleh, M., Hussain, M., Aboalsamh, H.: Breast mass classification using diverse contextual information and convolutional neural network. *Biosensors* **11**(11), 419 (2021). <https://doi.org/10.3390/bios11110419>
7. Chuang, Y.H., Huang, S.H., Hung, T.M., et al.: Convolutional neural network for human cancer types prediction by integrating protein interaction networks and omics data. *Sci. Rep.* **11**(1), 20691 (2021). <https://doi.org/10.1038/s41598-021-98814-y>
8. Hasan, N., Mishra, A., Arunkumar, R.: Fuzzy logic based cross-layer design to improve quality of service in mobile ad-hoc networks for next-gen cyber physical system. *Eng. Sci. Technol.* **35**, 101099 (2022). <https://doi.org/10.1016/j.jestch.2022.101099>
9. Hou, J., Aerts, J., den Hamer, B., et al.: Gene expression-based classification of non-small cell lung carcinomas and survival prediction. *PLoS ONE* **5**, e10312 (2010). <https://doi.org/10.1371/journal.pone.0010312>
10. Hu, Y., Zhao, L., Li, Z., et al.: Classifying the multi-omics data of gastric cancer using a deep feature selection method. *Expert Syst. Appl.* **200**, 116813 (2022). <https://doi.org/10.1016/j.eswa.2022.116813>
11. Li, J., Sun, W., Feng, X., et al.: A dense connection encoding-decoding convolutional neural network structure for semantic segmentation of thymoma. *Neurocomputing* **451**, 1–11 (2021). <https://doi.org/10.1016/j.neucom.2021.04.023>
12. Lytvynenko, V., Savina, N., Krejci, J., Voronenko, M., Yakobchuk, M., Kryvoruchko, O.: Bayesian networks' development based on noisy-max nodes for modeling investment processes in transport. In: *CEUR Workshop Proceedings*, vol. 2386 (2019)
13. Marasanov, V.V., Sharko, A.V., Sharko, A.A.: Energy spectrum of acoustic emission signals in coupled continuous media. *J. Nano-and Elect. Phys.* **11**(3), 03028 (2019). [https://doi.org/10.21272/jnep.11\(3\).03028](https://doi.org/10.21272/jnep.11(3).03028) [https://doi.org/10.21272/jnep.11\(3\).03028](https://doi.org/10.21272/jnep.11(3).03028) [https://doi.org/10.21272/jnep.11\(3\).03028](https://doi.org/10.21272/jnep.11(3).03028)
14. Morandat, F., Hill, B., Osvald, L., Vitek, J.: Evaluating the design of the R language. *Lect. Notes Comput. Sci.* **7313**, 104–131 (2012). https://doi.org/10.1007/978-3-642-31057-7_6
15. Qi, Y., Su, B., Lin, X., Zhou, H.: A new feature selection method based on feature distinguishing ability and network influence. *J. Biomed. Inf.* **128**, 104048 (2022). <https://doi.org/10.1016/j.jbi.2022.104048>
16. Sharko, M.V., Sharko, O.V.: Innovation aspects of enterprises development management in regional tourism. *Actual Prob. Econ.* **181**(7), 206–213 (2016)
17. Taherkhani, N., Sepehri, M., Khasha, R., Shafaghi, S.: Ranking patients on the kidney transplant waiting list based on fuzzy inference system. *BMC Nephrol.* **23**(1), 31 (2022). <https://doi.org/10.1186/s12882-022-02662-5>
18. Yuan, X., Liebelt, M., Shi, P., Phillips, B.: Cognitive decisions based on a rule-based fuzzy system. *Inf. Sci.* **600**, 323–341 (2022). <https://doi.org/10.1016/j.ins.2022.03.089>
19. Zadeh, L.: Fuzzy logic. *Computational Complexity: Theory, Techniques, and Applications*, pp. 1177–1200 (2013). https://doi.org/10.1007/978-1-4614-1800-9_73



Mathematical Model of Preparing Process of Bulk Cargo for Transportation by Vessel

Oksana Polyvoda¹(✉)  and Vladyslav Polyvoda² 

¹ Kherson National Technical University, Kherson, Ukraine
pov81@ukr.net

² Kherson State Maritime Academy, Kherson, Ukraine

Abstract. The aim of this study is to solve the problem of bulk cargo preparation by bringing its parameters, namely temperature and moisture to safe standard conditions for maritime transport by developing a mathematical model of drying, which will improve drying equipment control methods and optimize drying parameters to ensure energy-saving modes of operation of technological equipment. It is determined that the largest volume of bulk cargo is wheat grain among the bulk cargoes transported by vessel. Safe transportation of grain in bulk mainly depends on the conditional quality of grain at the time of loading on the ship, which is achieved by pre-drying the grain at the port facilities.

In this study, a continuous mathematical model of the grain drying process is obtained, based on heat and mass transfer equations, which makes it possible to predict changes in grain moisture taking into account the influence of thermophysical and thermodynamic properties of grain.

Based on a continuous mathematical model, a discretized mathematical model using the finite difference method and a model of the dynamics of the drying process in the state space using the linearization method by decomposing a nonlinear model into a Taylor series was developed and then was implemented in Mathcad. The experimental study was performed on research equipment, which showed the dependence of temperature and moisture of grain on time during drying. Analysis of the calculated errors of the mathematical model proves its adequacy, which allows its further use to optimize the drying process.

Keywords: Bulk cargo · Grain drying · Mathematical model · Finite difference method · State space

1 Introduction

For more than 100 years some cargoes such as grain and coal have been shipped around the world in bulk. The range of cargoes carried in bulk has increased dramatically during recent decades, rising to over 3 billion tonnes a year. The types of bulk cargoes carried - in addition to grain and coal - now include animal

foodstuffs, fertilizers, mineral concentrates and ores, scrap metal and biofuels such as wood pellets. The vast majority of bulk cargo voyages are completed without incident but there have been many occasions when things have not gone to plan. In some instances the cargo suffered damage but there was no harm done to the ship or the crew. On other occasions the ship suffered damage as a result of the cargo shifting. Unfortunately some ships have been lost with all hands because of problems with the cargo or because of defects or damage to the ship [2].

The International Maritime Organization (IMO) [9] publishes the International Convention for the Safety of Life at Sea (SOLAS) [19], codes of safe practice and guidelines that set out requirements that must be followed and complied with. Additionally, there are a number of books that give recommendations and guidance for the stowage and securing of particular items of cargo. SOLAS - is one of the principal conventions of the IMO, which periodically discusses and adopts requirements and amendments to it. The SOLAS Convention deals with various aspects of maritime safety and contains, in chapter VI, the mandatory provisions governing the carriage of solid bulk cargoes. These provisions are extended in the International Maritime Solid Bulk Cargoes Code (IMSBC Code) [10]. It is worth noting that the IMSBC Code does not cover the carriage of grain in bulk. The specific requirements for the transport of grain are covered by the International Code for the Safe Carriage of Grain in Bulk (International Grain Code, 1991) [21].

Therefore, an important task of this study is to solve the problem of preparation of bulk cargo by bringing their parameters, namely temperature and moisture to standard conditions for safe transport by vessel by developing a mathematical model of drying, which will improve drying equipment control and optimize drying parameters to ensure energy-saving modes of operation of port terminal equipment.

2 Problem Statement

The cereal grains are a major source of our dietary energy and protein requirements. They are also needed for animal feeding and industrial processing. There are three major cereal species according to volume of production: wheat, maize and rice. These three represent nearly 90% of cereal-grain production; 700–1000 million tonnes of each are produced annually [18].

From the time the goods are presented for transportation, they enter a new stage – they become cargo, and a number of new tasks arise for maritime transport, among which one of the most important ones stands out – the task of ensuring the safety of the consumer characteristics of the cargo, taking into account its transport properties. The transport properties of cargo are a set of cargo characteristics that determine the method and conditions of its transportation, processing and storage.

Depending on the influence of the external environment on the cargo, i.e. temperature and moisture, the cargoes are divided according to the mode of

transportation into non-regime and regime. Non-regime cargoes include such cargoes, in which the actions of aggressive factors arising in the process of sea transportation do not cause changes in their physical and chemical properties and deterioration in quality. They do not require the creation of special conditions during their transportation. Regime cargo includes such cargo that requires the creation of certain temperature and humidity conditions in the holds, as well as preliminary preparation for transportation. Without compliance with these conditions, the transportation of regime cargoes is impossible or only possible for a limited time. Some cargoes require special quarantine transportation regimes.

According to their physical and chemical properties, cargoes are divided into two main groups: unpreserved and stably preserved. In addition, almost all cargoes have specific, inherent properties that determine the requirements that must be met in the process of their transportation by sea.

The main parameters of grain cargoes transported in bulk include granulometric composition, moisture and temperature, which determine the angle of repose (or angle of rest), shrinkage, flowability, specific volume, self-heating, spontaneous combustion, caking of the cargo [3].

The wheat grain is one of the most produced cereals in the world is also one of the most difficult and dangerous to transport.

Ensuring the required and safe transport properties of wheat grain could be achieved both by preliminary bringing its quality to the required values before loading onto a ship, and by monitoring quality indicators, carrying out ventilation during transportation.

The main parameters controlled during the conditioning of grain before loading are temperature, moisture and ventilation conditions. There is no lower temperature limit and the favourable travel temperature is around 20 °C.

At temperatures over 25 °C the metabolic processes in grain mass increase, leading to increased CO₂ production and self-heating of the wheat.

With high moisture content of the grain, its baking properties are lost and the product will no longer be suitable for the production of flour. Moisture problems can be prevented by bringing the grain quality up to standard values by pre-drying the wheat.

Wheat grain moisture content of over 16–17% can rapidly produce an excessively damp atmosphere within the hold and may cause considerable damage to the cargo. In such conditions it could be possible that the self-heating clusters in grain mass are arisen and it following expansions at boundary layers.

At moisture levels of over 17%, swelling of the wheat grain in the hold occurs, what may result in structural damage to the ship.

Thus, the normal moisture content of export grain is 11–14%. Grain with a moisture content of 16% is prohibited for transportation.

Prior to loading, the moisture content should be checked by an independent inspector and a certificate provided. The certificates should state not only that appropriate measures have been carried out but also how and with what they were carried out and at what level of success [6].

Bringing grain quality up to standard values before transportation is one of the tasks of modern sea port terminals for grain shipment. To solve this problem, grain drying complexes are used. In principle, these are grain dryers of any type (mine, conveyor, drum, chamber, combined, etc.) together with all auxiliary and related equipment: loading and unloading bunkers, norias, conveyors and grain cleaning equipment. Such complexes are complete technological units, fully perform the task of drying and allow some time to store grain before transportation. Advantages of a conveyor (horizontal belt) grain dryer: gentle drying at low temperature, which makes it possible to use dried grain as a seed, energy efficiency, environmental friendliness, high operational availability, the presence of “smart” control.

3 Literature Review

To solve a set of practical problems, such as scientific reasoning of rational modes of technological processes, determination of temperature gradients and moisture content requires a mathematical model to determine the thermophysical and mass transfer characteristics of grain during drying using a conveyor grain dryer.

Researchers have developed various methods and approaches to analyze the dynamics of grain mass parameters during the drying process. The logarithmic models are based on a unique relationship between the drying rate and the temperature gradient with respect to depth. The main advantage of these models is their simplicity. However, they are not very accurate [5]. Usually, to study the thermophysical and mass transfer processes taking place in the grain mass, an analytical solution of the equations of heat and mass transfer is used [15]. The analytical solution of the corresponding simultaneous heat and mass transfer model could be obtained by introducing an auxiliary function through the variable separation approach [20]. In [8] was proposed an inverse algorithm based on the conjugate gradient method and the discrepancy principle. However, the practical application of such methods faces the problems of algorithmization in the development of final control systems. Recently, models and methods based on the use of fuzzy logic [11] and neural networks [16] have been developed. The disadvantage of using these models is that a large array of experimental data or preliminary observations is required.

Considering the disadvantages of existing models described above, to model the process of drying the grain mass, it is advisable to use the approach using a linearized model in the space of states, which allows to analyze grain moisture at the desired point of the grain layer [1, 14].

The objective of the research is to develop a discretized linear mathematical model of grain drying in space based on heat and mass transfer equations, which will improve the automated control system of grain drying process while bringing grain quality to standard conditions for safe transportation by sea.

4 Materials and Methods

In automated control systems of conveyor dryers, the selection and optimization of drying parameters, solution of control and process control tasks is performed taking into account data on grain temperature and moisture at each point of the grain layer. To determine the dynamics of these grain parameters traditionally used differential equations of heat and mass transfer [9], in the following form:

$$\frac{\partial W}{\partial \tau} = \frac{\partial}{\partial z} \left(a_m \frac{\partial W}{\partial z} + a_m \delta \frac{\partial t_g}{\partial z} \right), \quad (1)$$

$$c\rho_0 \frac{\partial t_g}{\partial \tau} = \frac{\partial}{\partial z} \left(\lambda \frac{\partial t_g}{\partial z} + \varepsilon r \rho_0 \frac{\partial W}{\partial z} \right), \quad (2)$$

where: W is the moisture; t_g is the temperature; ρ_0 is the density; c is the heat capacity; λ is the thermal conductivity of grain; a_m is the moisture diffusion coefficient; δ is the thermogradient coefficient; ε is the phase transformation criterion; z is the coordinate; τ is the time.

The initial conditions for Eqs. (1), (2) can be written as follows:

$$t_g(z, \tau)|_{t=\tau_0} = t_g(z), \quad W(z, \tau)|_{t=\tau_0} = W(z), \quad (3)$$

where $t_g(z)$ and $W(z)$ are the known functions.

Boundary conditions on the surface of the grain mass are determined primarily by the principle of distribution of the coolant, the initial moisture and temperature of the grain, the temperature of the drying agent, the intensity of heat flow. For example, for a cross-flow conveyor dryer in which the grain mass and the drying agent flows are directed perpendicularly, the boundary conditions can be set as

$$\frac{\partial t_g(z, \tau)}{\partial z} + \frac{\alpha(W(z, \tau))}{\lambda(W(z, \tau))} [T_{da} - t_g(z, \tau)] = 0, \quad (4)$$

$$\frac{\partial W(z, \tau)}{\partial z} + \frac{b \cdot t_g(z, \tau)}{a_m(W, t_g)} [W(z, \tau) - W_p] = 0, \quad (5)$$

where: empirical coefficient $b = 0.86 \cdot 10^{-7}$ for a dense layer; W_p is the equilibrium moisture content of grain ($W_p = 9.1\%$); T_{da} is the temperature of the drying agent for the upper limit conditions or the temperature due to the heating of the conveyor belt for the lower limit conditions.

Equations (1)–(5) contain partial derivatives, which makes it difficult to obtain an accurate solution, so a model of grain temperature and moisture dynamics was developed based on heat and mass transfer equations using the finite difference method [6], which allows predict the moisture $W(z, \tau)$ and temperature of the grain mass $t_g(z, \tau)$ in each layer of the grain mass as follows:

$$W(i, j+1) = a_m \frac{K}{h^2} [W(i+1, j) - 2W(i, j) + W(i-1, j)] + a_m \delta \frac{K}{h^2} [t_g(i+1, j) - 2t_g(i, j) + t_g(i-1, j)] + W(i, j), \quad (6)$$

$$t_g(i, j+1) = \frac{\lambda \cdot K}{c \cdot \rho_0 \cdot h^2} [t_g(i+1, j) - 2t_g(i, j) + t_g(i-1, j)] + \frac{\varepsilon \cdot r \cdot K}{c \cdot h^2} [W(i+1, j) - 2W(i, j) + W(i-1, j)] + t_g(i, j), \quad (7)$$

where: $i = 0, 1, \dots, N$; $j = 0, 1, \dots, M$; $K = t_c/M$; $h = l_g/N$; l_g is the height of the grain layer on the dryer belt; t_c is the time of the drying cycle; N is the number of nodal points in the height of the grain layer and M is the in time.

In solving Eqs. (1)–(5) an important issue is the formulation of initial and boundary conditions, which are determined taking into account the design and technological characteristics of grain dryers, and significantly affect the adequacy of the overall model of drying process dynamics.

After discretization of the initial (3) and boundary (4), (5) conditions by the finite difference method, the values of W_{i0} and $t_{g\ i0}$ on the left side of the grid are determined from the initial conditions as $W_{i0} = W(i, 0) = W_0$ and $t_{g\ i0} = t_g(i, 0) = t_{g\ 0}$ respectively. The temperature and moisture limits on the upper ($i = 0$) and lower ($i = N$) sides of the grid are calculated as follows:

$$t_g(0, j+1) = \frac{t_g(1, j+1) + \frac{h \cdot \alpha(i, j) \cdot T_{da}}{\lambda(i, j)}}{1 + \frac{h \cdot \alpha(i, j)}{\lambda(i, j)}}, \quad (8)$$

$$W(0, j+1) = \frac{-b \cdot t_g(1, j+1) \cdot h}{a_m(i, j)} (W(1, j+1) - W_p) + W(1, j+1). \quad (9)$$

To ensure the grain drying process, it is necessary to control the grain drying equipment taking into account the dynamics of moisture and grain temperature along the grain layer, which is possible using a discretized mathematical model (6)–(9), but the use of modern control methods implies the presence of an object model in the state space [7]. To do this, in the discretized equation of heat and mass transfer it is necessary to replace the variables:

$$t_g(0, j) = t_0, \quad t_g(1, j) = t_1 \dots t_g(N, j) = t_N, \\ W(0, j) = W_0, \quad W(1, j) = W_1 \dots W(N, j) = W_N, \quad T_{da} = U.$$

Then the model of the dynamics of temperature and moisture of the grain can be represented as follows:

$$\left\{ \begin{array}{l} \frac{dt_0}{d\tau} = \frac{\alpha(W_1)}{\lambda(W_1)} t_0 + \frac{-\alpha(W_1)}{\lambda(W_1)} U, \\ \frac{dt_i}{d\tau} = \frac{\lambda(W_i)}{c(W_i)\rho_0 h^2} [t_{i+1} - 2t_i + t_{i-1}] + \frac{\varepsilon r}{c(W_i)h^2} [W_{i+1} - 2W_i + W_{i-1}], \\ \frac{dt_N}{d\tau} = \frac{\alpha(W_N)}{\lambda(W_N)} t_{N-1}, \end{array} \right. \quad (10)$$

$$\left\{ \begin{array}{l} \frac{dW_0}{d\tau} = \frac{b}{a_m(W_1, t_1)} t_1 W_1 - \frac{b \cdot W_p}{a_m(W_1, t_1)} t_1, \\ \frac{dW_i}{d\tau} = \frac{a_m(W_i, t_i)}{h^2} [W_{i+1} - 2W_i + W_{i-1}] + \frac{a_m(W_i, t_i) \cdot \delta}{h^2} [t_{i+1} - 2t_i + t_{i-1}], \\ \frac{dW_N}{d\tau} = \frac{-b}{a_m(W_{N-1}, t_{N-1})} t_{N-1} \cdot W_{N-1} + \frac{b \cdot W_p}{a_m(W_{N-1}, t_{N-1})} t_{N-1}, \end{array} \right. \quad (11)$$

where: $W = (W_0, W_1, \dots, W_k, \dots, W_N)^T$ is the state vector (values of grain moisture by the height of the grain layer); $t = (t_0, t_1, \dots, t_k, \dots, t_N)^T$ is the state vector (values of grain temperature by the height of the grain layer).

Equations (10)–(11) can be represented in a simplified form:

$$\begin{cases} \dot{t}_0 = \varphi_0(W_1, t_0) + b_0(W_1)U, \\ \dot{t}_i = \varphi_i(W_{i-1}, W_i, W_{i+1}, t_{i-1}, t_i, t_{i+1}) + b_0(W_1)U, \\ \dot{t}_N = \varphi_N(W_N, t_{N-1}). \end{cases} \quad (12)$$

$$\begin{cases} \dot{W}_0 = \psi_0(W_1, t_1), \\ \dot{W}_i = \psi_i(W_{i-1}, W_i, W_{i+1}, t_{i-1}, t_i, t_{i+1}) + b_0(W_1)U, \\ \dot{W}_N = \psi_N(W_{N-1}, t_{N-1}). \end{cases} \quad (13)$$

where: $\varphi(W, t)$, $b(W)$ and $\psi(W, t)$ are the continuous functions depending on the elements of state vectors formed on the basis of thermophysical and thermodynamic characteristics of grain; U is the control vector formed taking into account the type of method of drying agent supply.

The restrictions on phase vectors are $W = (W_0, W_1, \dots, W_k, \dots, W_N)^T$, $t = (t_0, t_1, \dots, t_k, \dots, t_N)^T$ and control vector \mathbf{u} have the appearance

$$T_{env} \leq U \leq T_{\max da}, \quad T_0 \leq T_{\max g}, \quad (14)$$

where: T_{env} is the environmental temperature; $T_{\max da}$ is the maximum possible temperature of the drying agent for a specific type of drying equipment; $T_{\max g}$ is the maximum allowable grain heating temperature.

Boundary conditions can be written as follows:

$$t(\tau_0) = t_0 = const, \quad W(\tau_0) = W_0 = const, \quad (15)$$

where: $t(\tau_0)$, $W(\tau_0)$ corresponds to the initial temperature and moisture in the grain layer.

The obtained model of moisture dynamics is nonlinear. The simplest and most commonly used technique in modeling processes occurring in nonlinear multidimensional objects [4] is that the equations of dynamics are linearized in the surrounding area of some base solution [12].

Let us linearize the model of moisture dynamics (13) relatively to some steady state $t = (t_{0s}, t_{1s}, \dots, t_{Ns})^T$, $W = (W_{0s}, W_{1s}, \dots, W_{Ns})^T$ and specified controls \mathbf{u}_s . To do this, let's move on to increments $\hat{t}_i = t_i - t_{is}$, $\hat{W}_i = W_i - W_{is}$, $\hat{U} = U - U_g$ and convert equation (10) to the Taylor series, excluding terms above the first order [13,17]. As a result of transformations of the system (11)–(12) will look like:

$$\left\{ \begin{array}{l} \frac{\partial \hat{t}_0}{\partial \tau} = \frac{\alpha(W_{1s})}{\lambda(W_{1s})} \cdot \hat{t}_0 + (t_{0s} - U_s) \frac{\partial \left[\frac{\alpha(W_{1s})}{\lambda(W_{1s})} \right]}{\partial W_{1s}} \cdot \hat{W}_1 + \frac{-\alpha(W_{1s})}{\lambda(W_{1s})} \cdot \hat{U}, \\ \frac{\partial \hat{t}_i}{\partial \tau} = \frac{1}{\rho_0 h^2} \left[\frac{\lambda(W_{is})}{c(W_{is})} \cdot \hat{t}_{i+1} - \frac{2\lambda(W_{is})}{c(W_{is})} \cdot \hat{t}_i + \frac{\lambda(W_{is})}{c(W_{is})} \cdot \hat{t}_{i-1} \right] \\ + \frac{\varepsilon r}{h^2} \left[\frac{1}{c(W_{is})} \cdot \hat{W}_{i+1} + \frac{1}{c(W_{is})} \cdot \hat{W}_{i-1} \right] \\ + \left[\frac{t_{(i+1)s} - 2t_{is} + t_{(i-1)s}}{\rho_0 h^2} \cdot \frac{\partial \left[\frac{\lambda(W_{is})}{c(W_{is})} \right]}{\partial W_{is}} - \frac{2\varepsilon r}{h^2} \cdot \frac{\partial \left[\frac{\lambda(W_{is})}{c(W_{is})} \right]}{\partial W_{is}} \right] \cdot \hat{W}_i, \\ \frac{\partial \hat{t}_N}{\partial \tau} = \frac{\alpha(W_{Ns})}{\lambda(W_{Ns})} \hat{t}_{N-1} + t_{(N-1)s} \frac{\partial \left[\frac{\alpha(W_{Ns})}{\lambda(W_{Ns})} \right]}{\partial W_{Ns}} \cdot \hat{W}_N, \end{array} \right. \quad (16)$$

$$\left\{ \begin{array}{l} \frac{\partial \hat{W}_0}{\partial \tau} = \frac{b(W_{1s}, W_p)}{a_m(W_{1s}, t_{1s})} \cdot \frac{\partial \left(\frac{1}{a_m(W_{1s}, t_{1s})} \right)}{\partial t_{1s}} \cdot \hat{t}_1 + b \cdot t_{1s} \cdot \frac{\partial \left(\frac{1}{a_m(W_{1s}, t_{1s})} \right)}{\partial W_{1s}} \\ \times \left(\frac{1}{a_m(W_{1s}, t_{1s})} + W_p \right) \cdot \hat{W}_1, \\ \frac{\partial \hat{W}_i}{\partial \tau} = \frac{1}{h^2} \left[a_m(W_{is}, t_{is}) \cdot \hat{W}_{i+1} + \frac{\partial a_m(W_{is}, t_{is})}{\partial W_{is}} (-2a_m(W_{is}, t_{is}) \right. \\ + \delta(t_{(i-1)g} - 2t_{ig} + t_{(i+1)g})) \times \hat{W}_i + a_m(W_{is}, t_{is}) \cdot \hat{W}_{i-1} \\ + \delta \cdot a_m(W_{is}, t_{is}) \cdot \hat{t}_{i+1} + \frac{\partial a_m(W_{is}, t_{is})}{\partial t_{is}} \\ \times (-2\delta \cdot a_m(W_{is}, t_{is}) + W_{(i+1)s} - 2W_{is} + t_{(i-1)s}) \cdot \hat{t}_i \\ \left. + a_m(W_{is}, t_{is}) \delta \cdot \hat{t}_{i-1} \right], \\ \frac{\partial \hat{W}_N}{\partial \tau} = \frac{b(W_p - W_{(N-1)s})}{a_m(W_{(N-1)s}, t_{(N-1)s})} \cdot \frac{\delta \left(\frac{1}{a_m(W_{(N-1)s}, t_{(N-1)s})} \right)}{\partial t_{(N-1)s}} \cdot \hat{t}_{N-1} \\ + b \cdot t_{(N-1)s} \cdot \frac{\delta \left(\frac{1}{a_m(W_{(N-1)s}, t_{(N-1)s})} \right)}{\partial W_{(N-1)s}} \\ \times \left(\frac{1}{a_m(W_{(N-1)s}, t_{(N-1)s})} + W_p \right) \cdot \hat{W}_{(N-1)}. \end{array} \right. \quad (17)$$

When determining partial derivatives, we use analytical dependences for thermophysical and thermodynamic parameters of grain:

$$\begin{aligned} \alpha(W) &= a_\alpha W + b_\alpha, \\ \lambda(W) &= a_\lambda W + b_\lambda, \end{aligned} \quad (18)$$

$$c(W) = \begin{cases} a_{c1} W + b_{c1}, & \text{if } 1.8\% \leq W < 7.7\%, \\ a_{c1} W + b_{c1}, & \text{if } 7.7\% \leq W < 23.7\%, \\ a_{c1} W + b_{c1}, & \text{if } 23.7\% \leq W < 33.6\%, \end{cases} \quad (19)$$

$$a_m(W, t) = \begin{cases} b_{am1} W + c_{am1}, & \text{if } 10\% \leq W < 21\%, \\ \frac{\left(\frac{t}{T_0}\right)^{d_{am2} W + e_{am2}}}{a_{am2} W^2 + b_{am2} W + c_{am2}}, & \text{if } 21\% \leq W < 24\%, \\ \frac{\left(\frac{t}{T_0}\right)^{d_{am3} W + e_{am3}}}{a_{am3} W^2 + b_{am3} W + c_{am3}}, & \text{if } 24\% \leq W < 26.7\%, \\ b_{am4} W + c_{am4} \left(\frac{t}{T_0}\right)^{d_{am4} W + e_{am4}}, & \text{if } 26.7\% \leq W < 32\%, \end{cases} \quad (20)$$

where: $a_\lambda = 0.06$, $b_\lambda = 0.002$; $a_\alpha = 2.5$, $b_\alpha = 0.05$; $a_{ci} = (0.297, 0.263, 0.336)$, $b_{ci} = (5.91, 1.036, 7.31) \times 10^{-3}$, $i = \overline{1, 3}$; $a_{amj} = (0, 0.0558, 0.362, 0) \times 10^{-5}$, $b_{amj} = (-0.2, -382, 2340, 0.055) \times 10^{-9}$, $c_{amj} = (0.55, 67.8, 380.4, 0.147) \times 10^{-9}$, $d_{amj} = (0, 179, 16.5, 16.5)$, $e_{amj} = (0, -41.5, 11.72, 11.72)$, $j = \overline{1, 4}$.

For example, for grain moisture in the range [7.7%, 21%] we obtain linearized models of temperature dynamics in the form:

$$\left\{ \begin{array}{l} \frac{\partial \hat{t}_0}{\partial \tau} = \frac{a_\alpha W_{1s} + b_\alpha}{a_\lambda W_{1s} + b_\lambda} \hat{t}_0 + \frac{(a_\alpha b_\lambda - a_\lambda b_\alpha)(t_{0s} - U_g)}{(a_\lambda W_{1s} + b_\lambda)^2} \hat{W}_1 - \frac{a_\alpha W_{1s} + b_\alpha}{a_\lambda W_{1s} + b_\lambda} \hat{U}, \\ \frac{\partial \hat{t}_i}{\partial \tau} = \frac{1}{\rho_0 h^2} \left[\frac{a_\lambda W_{is} + b_\lambda}{a_{c2} W_{is} + b_{c2}} \cdot \hat{t}_{i+1} - \frac{2(a_\lambda W_{is} + b_\lambda)}{a_{c2} W_{is} + b_{c2}} \cdot \hat{t}_i + \frac{a_\lambda W_{is} + b_\lambda}{a_{c2} W_{is} + b_{c2}} \cdot \hat{t}_{i-1} \right] \\ \quad + \frac{\varepsilon r^2}{h^2} \cdot \frac{1}{a_{c2} W_{is} + b_{c2}} (\hat{W}_{i+1} + \hat{W}_{i-1}) + \frac{1}{h^2 (a_{c2} W_{is} + b_{c2})^2} \\ \quad \times \left[\frac{t_{(i+1)s} - 2t_{is} + t_{(i-1)s}}{\rho_0} \cdot (a_\lambda b_{c2} - a_{c2} b_\lambda) - 2\varepsilon r \cdot b_{c2} \right] \cdot \hat{W}_i, \\ \frac{\partial \hat{t}_N}{\partial \tau} = \frac{a_\alpha W_{Ns} + b_\alpha}{a_\lambda W_{Ns} + b_\lambda} \hat{t}_{N-1} + t_{(N-1)s} \cdot \frac{a_\alpha b_\lambda - a_\lambda b_\alpha}{(a_\lambda W_{Ns} + b_\lambda)} \cdot \hat{W}_N. \end{array} \right. \quad (21)$$

For the values of grain moisture in the depth of the layer the following ratios are obtained:

$$\left\{ \begin{array}{l} \frac{\partial \hat{W}_0}{\partial \tau} = b \cdot t_{1s} \cdot \frac{-b_{am}}{(b_{am1} W_{1s} + c_{am1})^2} \cdot \left(\frac{1}{b_{am1} W_{1s} + c_{am1}} + W_p \right) \cdot \hat{W}_1, \\ \frac{\partial \hat{W}_s}{\partial \tau} = \frac{1}{h^2} \left[(b_{am1} W_{is} + c_{am1}) \hat{W}_{i+1} - \frac{b_{am1}}{(b_{am1} W_{is} + c_{am1})^2} \right. \\ \quad \times (-2(b_{am1} W_{is} + c_{am1}) + \delta(t_{(i+1)s} - 2t_{is} + t_{(i+2)s})) \hat{W}_i \\ \quad \left. + (b_{am1} W_{is} + c_{am1}) \hat{W}_{i-1} + \delta(b_{am1} W_{is} + c_{am1}) (\hat{t}_{i+1} + \hat{t}_{i-1}) \right], \\ \frac{\partial \hat{W}_N}{\partial \tau} = \frac{-b_{am1} b t_{(N-1)s}}{(b_{am1} W_{(N-1)s} + c_{am1})^2} \left[\frac{1}{b_{am1} W_{(N-1)s} + c_{am1}} + W_p \right] \hat{W}_{N-1}, \end{array} \right. \quad (22)$$

where: t_{0s} , t_{is} , t_{Ns} , W_{0s} , W_{is} , W_{1s} , W_{Ns} are the stable values of temperature and moisture of grain on the surface, at the depth of the layers and at the lower limit, respectively.

These stable values are defined as solutions of systems of Eqs. (10)–(11) with zero left side.

For grain temperature, the system of equations has the form:

$$\left\{ \begin{array}{l} 0 = \frac{a_\alpha W + b_\alpha}{a_\lambda W + b_\lambda} t_{0s} - \frac{a_\lambda W + b_\lambda}{a_\alpha W + b_\alpha} U_s, \\ 0 = \frac{a_\lambda W + b_\lambda}{a_{c2} W + b_{c2} \rho_0 h^2} [t_{(i+1)s} - 2t_{is} + t_{(i-1)s}] \\ \quad + \frac{\varepsilon r}{(a_{c2} W + b_{c2}) h^2} \cdot [W_{(i+1)s} - 2W_{is} + W_{(i-1)s}], \\ 0 = \frac{a_\alpha W + b_\alpha}{a_\lambda W + b_\lambda} t_{(N-1)s}, \end{array} \right. \quad (23)$$

and for moisture:

$$\begin{cases} 0 = \frac{b}{b_{am1}W+c_{am1}}t_{1s}W_{1s} - \frac{bW_p}{b_{am1}W+c_{am1}}t_{1s}, \\ 0 = \frac{(b_{am1}W+c_{am1})\delta}{h^2} \cdot [t_{(i-1)s} - 2t_{is} + t_{(i+1)s}] \\ + \frac{b_{am1}W+c_{am1}}{h^2} \cdot [W_{(i-1)s} - Wt_{is} + W_{(i+1)s}], \\ 0 = \frac{-b}{b_{am1}W+c_{am1}}t_{(N-1)s}W_{(N-1)s} + \frac{bW_p}{b_{am1}W+c_{am1}}t_{(N-1)s}. \end{cases} \quad (24)$$

Linearized equations (21)–(22) of the dynamics of temperature and moisture are written in vector-matrix form as follows:

$$\begin{bmatrix} \hat{\mathbf{t}} \\ \hat{\mathbf{W}} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \hat{\mathbf{t}} \\ \hat{\mathbf{W}} \end{bmatrix} + \mathbf{B}\hat{\mathbf{u}}, \quad (25)$$

where the matrix \mathbf{A} has a constant structure of the form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}.$$

The matrix \mathbf{A}_{11} has the form:

$$\mathbf{A}_{11} = \begin{bmatrix} a_{00} & 0 & \dots & 0 & \dots & 0 & 0 \\ a_{10} & a_{11} & a_{12} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & a_{i,j-1} & a_{i,j} & a_{i,j+1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{N-1,N-2} & a_{N-1,N-1} & a_{N-1,N} \\ 0 & \dots & 0 & \dots & 0 & a_{N,N-1} & 0 \end{bmatrix},$$

The matrix \mathbf{A}_{12} has the form $\mathbf{A}_{12} = [\mathbf{A}'_{12} \mathbf{A}''_{12}]$, where

$$\mathbf{A}'_{12} = \begin{bmatrix} 0 & a_{0,N+2} & \dots & 0 \\ a_{1,N+1} & a_{1,N+2} & a_{1,N+3} & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & a_{i,N+j-1} & a_{i,N+j} \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}''_{12} = \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ \dots & \dots & \dots \\ a_{i,N+j+1} & \dots & 0 \\ \dots & \dots & \dots \\ a_{N-1,2N-2} & a_{N-1,2N-1} & a_{N-1,2N} \\ 0 & \dots & a_{N,2N} \end{bmatrix}$$

The matrix \mathbf{A}_{21} has the form:

$$\mathbf{A}_{21} = \begin{bmatrix} 0 & \dots & 0 & \dots & \dots & 0 & 0 \\ a_{N+2,0} & 0 & a_{N+2,2} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & a_{N+i+1,j-1} & 0 & a_{N+i+1,j+1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{2N-1,N-2} & 0 & a_{2N-1,N} \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 \end{bmatrix}$$

The matrix \mathbf{A}_{22} has the form $\mathbf{A}_{22} = [\mathbf{A}'_{22} \mathbf{A}''_{22}]$, where:

$$\mathbf{A}'_{22} = \begin{bmatrix} 0 & a_{N+1,N+2} & 0 & \dots \\ a_{N+2,N+3} & a_{N+2,N+4} & a_{N+2,N+5} & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & a_{N+i+1,N+j-1} & a_{N+i+1,N+j} \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}''_{22} = \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ \dots & \dots & \dots \\ a_{N+i+1,N+j+1} & \dots & 0 \\ \dots & \dots & \dots \\ a_{2N-1,2N-2} & a_{2N-1,2N-1} & a_{2N-1,2N} \\ 0 & a_{2N-1,2N-1} & 0 \end{bmatrix}$$

The matrix \mathbf{B} has a constant structure of the form:

$$\mathbf{B}_1 = (b_{00} \ 0 \ \dots \ 0)^T.$$

Matrix \mathbf{A} is block. Matrix \mathbf{B} – with all zero rows except the first. The elements of the matrices are determined from the ratios:

$$\text{for } i = 0 : \\ b_{00} = -\frac{a_\alpha W_{1s} + b_\alpha}{a_\lambda W_{1s} + b_\lambda}, a_{00} = \frac{a_\alpha W_{1s} + b_\alpha}{a_\lambda W_{1s} + b_\lambda}, a_{0,N+2} = \frac{(a_\alpha b_\lambda - a_\lambda b_\alpha)(t_{0s} - U_g)}{(a_\lambda W_{1s} + b_\lambda)^2};$$

for $i = \overline{1, N-1}$:

$$a_{i,j-1} = \frac{1}{\rho_0 h^2} \cdot \frac{a_\lambda W_{is} + b_\lambda}{a_{c2} W_{is} + b_{c2}}, \quad a_{ij} = -\frac{1}{\rho_0 h^2} \cdot \frac{2(a_\lambda W_{is} + b_\lambda)}{a_{c2} W_{is} + b_{c2}}, \\ a_{i,i+1} = \frac{1}{\rho_0 h^2} \cdot \frac{a_\lambda W_{is} + b_\lambda}{a_{c2} W_{is} + b_{c2}}, \\ a_{i,N+j-1} = \frac{\varepsilon r^2}{h^2} \cdot \frac{1}{a_{c2} W_{is} + b_{c2}},$$

$$a_{i,N+j} = \frac{1}{h^2(a_{c2}W_{is} + b_{c2})^2} \left[\frac{t_{(i+1)s} - 2t_{is} + t_{(i-1)s}}{\rho_0} (a_\lambda b_{c2} - a_{c2}b_\lambda) - 2\epsilon r b_{c2} \right],$$

$$a_{i,N+j+1} = \frac{\epsilon r^2}{h^2} \cdot \frac{1}{a_{c2}W_{is} + b_{c2}};$$

for $i = N$:

$$a_{N,N-1} = \frac{a_\alpha W_{Ns} + b_\alpha}{a_\lambda W_{Ns} + b_\lambda},$$

$$a_{N,2N} = \frac{a_\alpha b_\lambda - a_\lambda b_\alpha}{(a_\lambda W_{Ns} + b_\lambda)^2},$$

for $i = N + 1$:

$$a_{N+1,N+2} = b \cdot t_{1s} \cdot \frac{-b_{am}}{(b_{am1}W_{1s} + c_{am1})^2} \cdot \left(\frac{1}{b_{am1}W_{1s} + c_{am1}} + W_p \right);$$

for $i = \overline{N + 2, 2N - 1}$:

$$a_{N+i+1,j-1} = \frac{1}{h^2} \cdot \delta(b_{am1}W_{is} + c_{am1}),$$

$$a_{N+i+1,j+1} = \frac{1}{h^2} \cdot \delta(b_{am1}W_{is} + c_{am1}),$$

$$a_{N+i+1,N+j-1} = \frac{1}{h^2} \cdot (b_{am1}W_{is} + c_{am1}),$$

$$a_{N+i+1,N+j+1} = \frac{1}{h^2} \cdot (b_{am1}W_{is} + c_{am1}),$$

$$a_{N+i+1,N+j} = \frac{1}{h^2} \cdot \frac{-b_{am1}}{(b_{am1}W_{is} + c_{am1})^2} \times (-2(b_{am1}W_{is} + c_{am1}) + \delta(t_{(i+1)s} - 2t_{is} + t_{(i+1)s})),$$

$$a_{2N,2N-1} = bt_{(N-1)s} \frac{-b_{am1}}{(b_{am1}W_{(N-1)s} + c_{am1})^2} \cdot \left[\frac{1}{b_{am1}W_{(N-1)s} + c_{am1}} + W_p \right].$$

Application of the received dependences gives the possibility to carry out modeling of work of the grain drying equipment and setup of mode parameters of drying process for any grain dryer of conveyor type.

5 Experiment and Results

The developed linearized mathematical model was implemented in the Mathcad software package. At the first stage (grain heating) the grain mass was studied with the following initial characteristics: $t_{g0} = 27.6$ °C, $W_0 = 0.22$ (22%). Drying cycle time $t_c = 60$ min, height of the grain layer on the dryer belt $l_g = 0.2$ m, the number of nodal points in the height of the grain layer $N = 10$, the number of nodal points in time $M = 3600$, the temperature of the drying agent $T_{da} = 100$ °C. In the modeling the quantitative characteristics of thermo-physical and thermodynamic properties of grain were used from the analytical dependences (18)–(20). The results of modeling of the temperature and moisture dynamics in the layers of the grain mass at the first stage of active drying are shown in Fig. 1. At the second stage (dry aeration in resting time) the grain mass was studied with the average characteristics obtained at the end of the first stage: $t_g = 58.7$ °C, $W = 0.213$ (21.3%), the resting time was $t_c = 60$ min. The results of modeling of the temperature and moisture dynamics in the layers of the grain mass in resting time are shown in Fig. 2. At the end of the resting time the average temperature was $t_g = 45.1$ °C and moisture $W = 0.207$ (20.7%).

To verify the adequacy of the developed mathematical model, a number of experimental studies were conducted. For experiments, a laboratory 28 liters drying oven SP-30 with forced convection was used. The material was placed in an experimental cell with a mesh base. Initial moisture content of grain 22%, drying time 60 min. The experiment was conducted indoors at stable environment conditions with wheat of the 6th class according to Ukrainian State Standard DSTU 3768-98. Samples were taken according to DSTU 13586.3-83. During drying, the grain temperature was monitored. Three DS18B20 sensors were installed to accurately monitor changes of the grain mass temperature during drying. Another sensor was located inside the oven to monitor the air temperature inside the chamber. After drying the grain, its moisture was measured using a “Wile 55” moisture meter. The temperature and moisture of the grain mass in the resting process were monitored according to a similar principle. Figure 3 shows the fragments of the calculated (solid line) and experimental (dotted line) dependences of the grain temperature in the middle layer by time in the first (Fig. 3, a) and second (Fig. 3, b) stages of the drying process. Figure 4 shows the results of calculations of the relative error for the two stages of drying.

6 Discussions

The resulting mathematical model of the temperature and moisture dynamics of the bulk grain layer makes it possible to perform the synthesis of automatic control systems for those channels through which automatic control is carried out. This model is universal for conveyer-type grain dryers with different design and

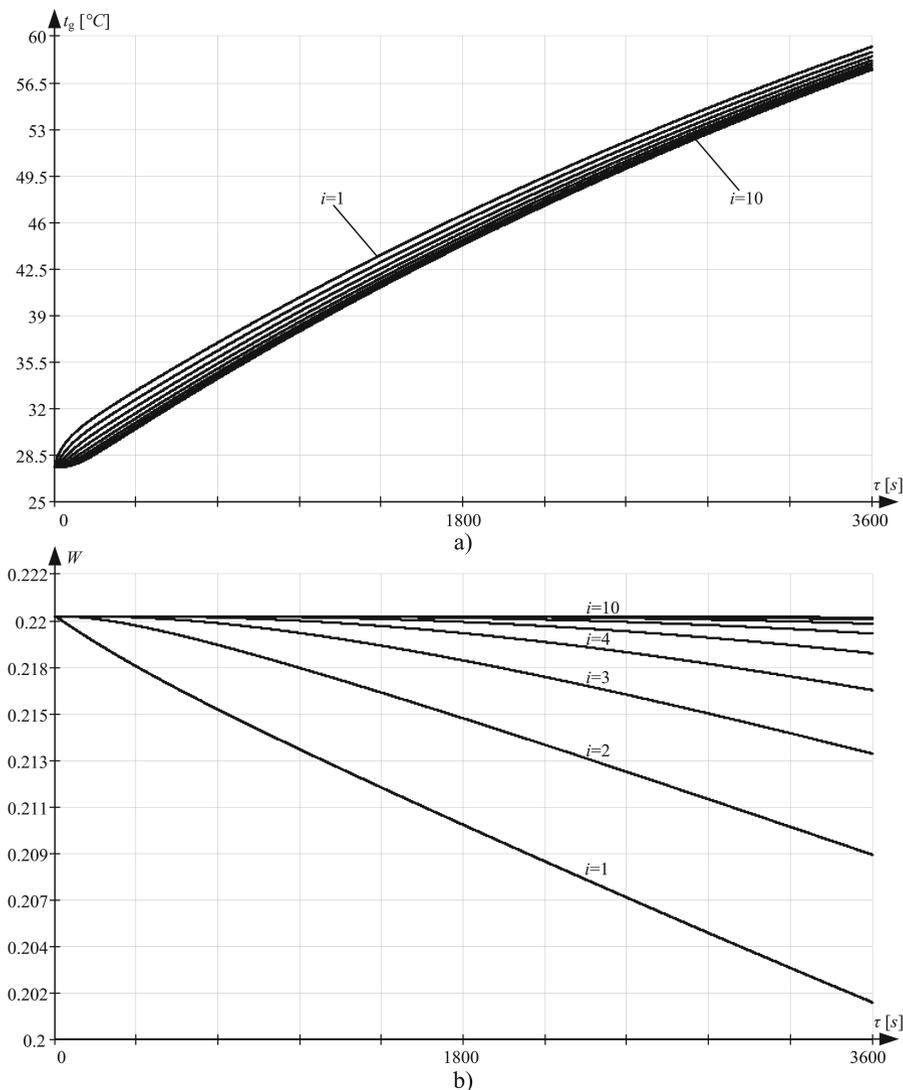


Fig. 1. Dynamics of temperature (a) and moisture (b) of the grain mass by layers at the active drying stage

technological schemes. It allows to analyze the dynamics of bulk grain parameters depending on the thickness of the grain layer, the temperature of the drying agent, the initial and current temperature of heating the grain, as well as the speed of the conveyor belt (an indirect characteristic of which is the duration of blowing the grain layer with the drying agent).

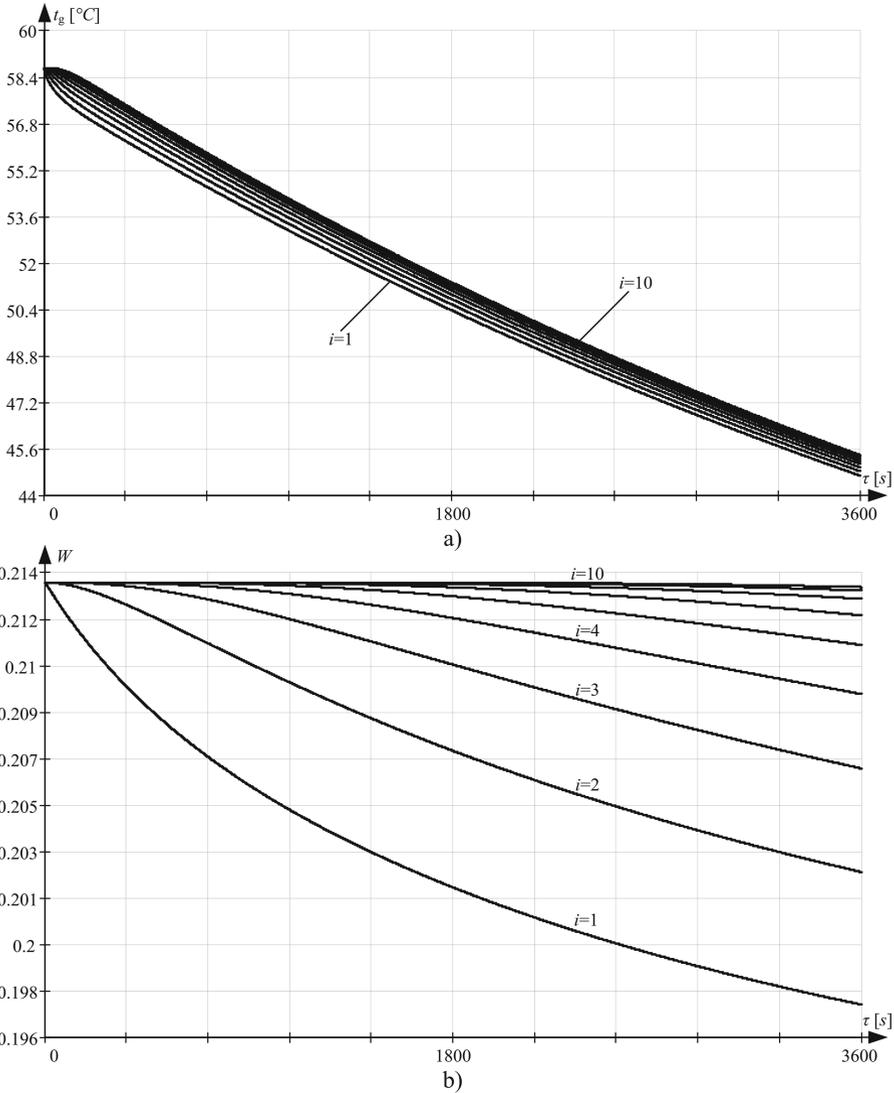


Fig. 2. Dynamics of temperature (a) and moisture (b) of the grain mass by layers at the resting time

In addition, the developed mathematical model of the drying process allows to predict the dynamics of temperature and moisture of the grain mass both at the stage of active drying with direct heating of heated air and “dry aeration”, when the decrease in moisture and grain temperature occurs at the expense of thermal energy.

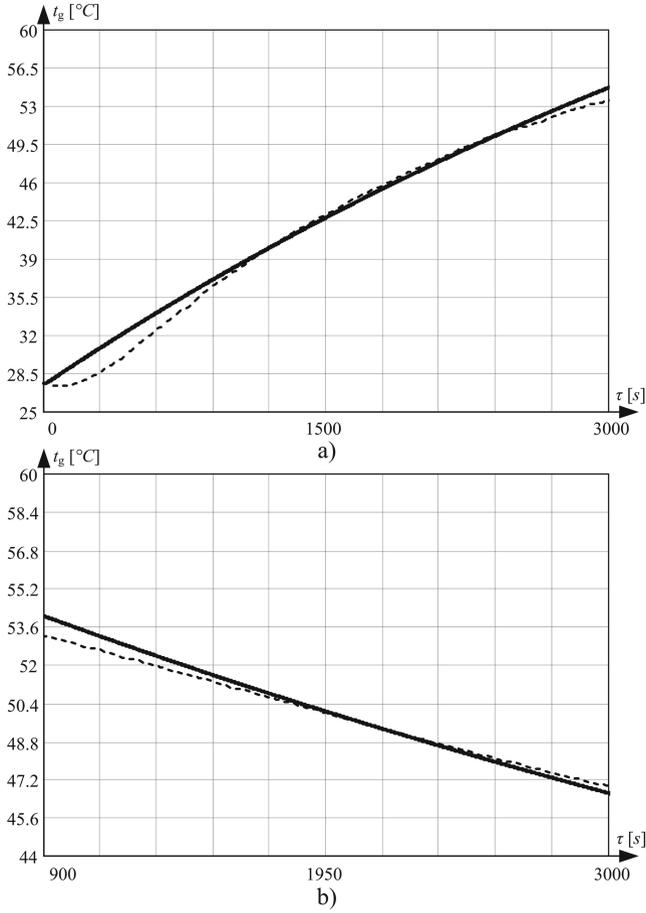


Fig. 3. Calculated (solid line) and experimental (dotted line) dependences of grain temperature by time: a) in the first stage, b) in the second stage

Analysis of the calculated errors of the mathematical model proves its adequacy, which allows its further use to improve the automated control system of the drying process to ensure energy-saving modes of operation of technological equipment.

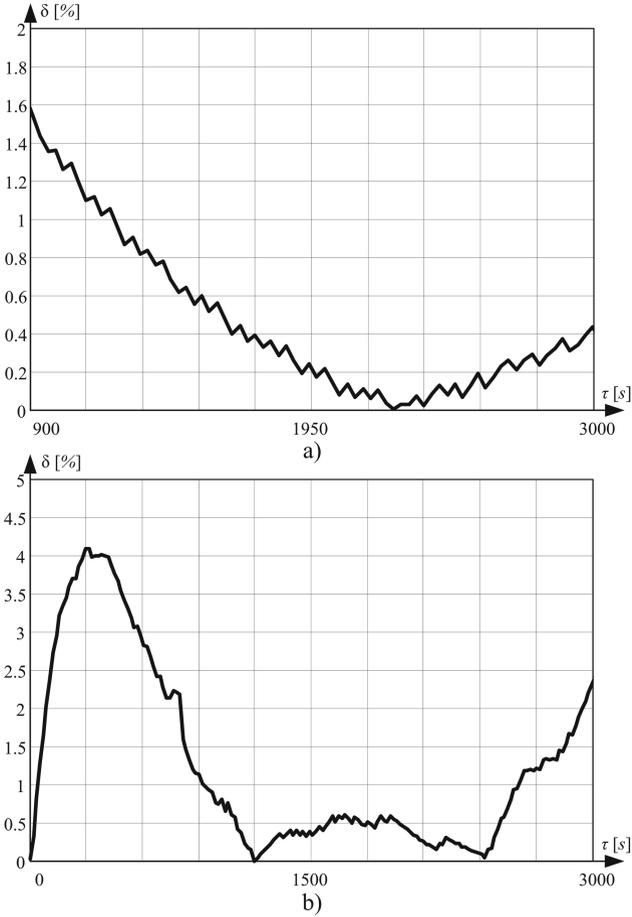


Fig. 4. Calculated dependences of model error by time: a) in the first stage, b) in the second stage

7 Conclusions

In this article, the problem of preparing bulk cargo for transportation by sea was solved, namely, bringing its parameters, temperature and moisture to safe standard conditions.

The analysis of international normative documents regulating the transportation of bulk cargo by sea is performed and the main parameters influencing the transportation process are highlighted.

The analysis of existing models, methods and means of control of grain drying equipment is performed. Necessity of development of model of dynamics of temperature and moisture of grain in the process of drying taking into account change of coefficients of heat exchange, diffusion of moisture, heat capacity, heat

conductivity that will allow to carry out optimization of mode parameters of control process of grain drying equipment is substantiated. Determination of thermophysical and mass transfer characteristics of grain is necessary to study the processes occurring during grain drying, knowledge of coefficients allows to scientifically substantiate rational modes of technological processes, which is important in practice, and allows to determine temperature and moisture content gradients and describe temperature and moisture content fields in grain, i.e. to obtain characteristics that affect its technological properties.

A discrete mathematical model of the grain drying process and a state space model based on heat and mass transfer equations have been developed, which makes it possible to predict the dynamics of grain temperature and moisture taking into account the influence of thermophysical and thermodynamic properties of grain. Analysis of the fundamental properties of the obtained system in the state space proved its complete controllability and observability. The received mathematical model of drying will allow to improve methods of control of the drying equipment and to optimize mode parameters of drying for the purpose of maintenance of energy-saving modes of functioning of the technological equipment.

References

1. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopolym. Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
2. Bliault, C., Jonas, M.: *Bulk Cargoes: A Guide to Good Practice*, p. 349. North of England P&I Club (2019)
3. Browne, C., Smith, P.L.: *Dry Cargo Chartering*, p. 320. Institute of Chartered Shipbrokers (2017)
4. Butkovskiy, A.G., Pustyl'nikov, L.M.: *Characteristics of Distributed-Parameter Systems: Handbook of Equations of Mathematical Physics and Distributed-Parameter Systems*. MAIA, Springer, Netherlands (2012). books.google.com.ua/books?id=lnX8CAAAQBAJ, <https://doi.org/10.1007/978-94-011-2062-3>
5. de C. Lopes, D., Neto, A.J.S., Santiago, J.K.: Comparison of equilibrium and logarithmic models for grain drying. *Biosyst. Eng.* **118**, 105–114 (2014). <https://doi.org/10.1016/j.biosystemseng.2013.11.011>
6. Chakraverty, S., Mahato, N., Karunakar, P., Rao, T.D.: *Advanced Numerical and Semi-Analytical Methods for Differential Equations*. Wiley (2019). books.google.com.ua/books?id=uiRDwAAQBAJ
7. Daras, N.J., Rassias, T.M.: *Modern Discrete Mathematics and Analysis: With Applications in Cryptography, Information Systems and Modeling*, SOIA, p. 521. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-74325-7>
8. Ebrahimifakhar, A., Yuill, D.: Inverse estimation of thermophysical properties and initial moisture content of cereal grains during deep-bed grain drying. *Biosyst. Eng.* **196**, 97–111 (2020). <https://doi.org/10.1016/j.biosystemseng.2020.05.021>
9. IMO: The international maritime organization. <https://www.imo.org/>
10. IMSBC: Imsbc code. www.imo.org/en/OurWork/Safety/Pages/BulkCarriers.aspx

11. Kaveh, M., Abbaspour-Gilandeh, Y., Chen, G.: Drying kinetic, quality, energy and exergy performance of hot air-rotary drum drying of green peas using adaptive neuro-fuzzy inference system. *Food Bioprod. Process.* **124**, 168–183 (2020). <https://doi.org/10.1016/j.fbp.2020.08.011>
12. Kogut, P.I., Kuppenko, O.P.: *Approximation Methods in Optimization of Nonlinear Systems*. De Gruyter Series in Nonlinear Analysis and Applications, De Gruyter (2019). <https://books.google.com.ua/books?id=loVaEAAAQBAJ>
13. Kondratenko, Y.P., Chikrii, A.A., Gubarev, V.F., Kacprzyk, J.: *Advanced Control Techniques in Complex Engineering Systems: Theory and Applications: Dedicated to Professor Vsevolod M. Kuntsevich*. SSDC, Springer, Cham (2019). books.google.com.ua/books?id=OxaaDwAAQBAJ, DOI: <https://doi.org/10.1007/978-3-030-21927-7>
14. Litvinenko, V.I., Burgher, J.A., Vyshemirskij, V.S., Sokolova, N.A.: Application of genetic algorithm for optimization gasoline fractions blending compounding. In: *Proceedings - 2002 IEEE International Conference on Artificial Intelligence Systems, ICAIS 2002*, pp. 391–394 (2002). <https://doi.org/10.1109/ICAIS.2002.1048134>
15. Luikov, A.V.: *Analytical Heat Diffusion Theory*. Academic Press, New York (1968). <https://doi.org/10.1016/B978-0-12-459756-3.X5001-9>
16. Nanvakenari, S., Movagharnjad, K., Latifi, A.: Evaluating the fluidized-bed drying of rice using response surface methodology and artificial neural network. *LWT* **147**, 111589 (2021). <https://doi.org/10.1016/j.lwt.2021.111589>
17. Rudakova, G., Polivoda, O.: Methods of the models building of UAV system in state space. In: *2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, pp. 139–144 (2015). <https://doi.org/10.1109/APUAVD.2015.7346584>
18. SKULD: Transportation of wheat (2017). <https://www.skuld.com/topics/cargo/solid-bulk/agricultural-cargoes/transportation-of-wheat/>
19. SOLAS. https://en.wikipedia.org/wiki/SOLAS_Convention
20. Wang, C., Wang, S., Jin, X., Zhang, T., Ma, Z.: Analytical solution for the heat and mass transfer of spherical grains during drying. *Biosyst. Eng.* **212**, 399–412 (2021). <https://doi.org/10.1016/j.biosystemseng.2021.11.006>
21. Wrigley, C., Batey, I., Miskelly, D.: *Cereal Grains: Assessing and Managing Quality*, p. 788. Woodhead Publishing (2017)



Computer Simulation of Joule-Thomson Effect Based on the Use of Real Gases

Vasiliy Nadraga¹ , Anatolii Balanda² , Mariana Polodiuk² ,
Yuliia Bobyr² , and Tetiana Kochura² 

¹ Educational and Scientific Institute of Management, Economy and Nature Management of V.I. Vernadsky Taurida National University, Kyiv, Ukraine
nadruga-vasil@ukr.net

² Military Academy named after Eugene Bereznyak, Kyiv, Ukraine
{anatol_ssu,mar_polod}@ukr.net

Abstract. Computer simulation of various nature processes can allow us to optimize the system parameters in order to increase the efficiency of its functioning in real conditions. In this paper, we present a model of a heating system based on the Joule-Thomson effect, the working fluid of which is a real gas. In order to compare the efficiency of heating systems, in which water and real gas are used as the working fluid, formulas for the efficiency of the corresponding models were obtained, which made it possible to optimize the parameters of the corresponding model. A computer simulation was carried out using the tools of R software. As a result of the simulation, the diagrams of the system parameters variation under various conditions of the model operating were obtained, which made it possible to optimize the parameters of the model in order to increase the efficiency of the system.

Keywords: Joule-Thomson effect · Computer simulation · Heating system · Optimization of system parameters · The system efficiency

1 Introduction

The use of computer simulation methods for the analysis of various nature processes when the creation of appropriate models and decision-making support systems is one of the topical areas of current applied science. An adequately implemented model can allow us not only to understand the nature of the investigated processes behavior, but also to optimize the system parameters in order to increase the efficiency of its functioning. In the general case, the mathematical model of the process underlying the operation of the studied system is represented as a function of the dependence of the model output parameters versus the input parameters. In this instance, the input parameters are divided into static (unchangeable) and dynamic (can be changed). The optimal operating mode of the model is determined by a certain combination of input parameters, which corresponds to the extreme of the given model operation quality

function. In the simplest case, the model can be represented analytically in the form of an appropriate function that determines the nature of the investigated dependencies. Graphical visualization of the required dependencies can allow us to optimize the model operation by searching of the obtained function global maximum or minimum. When analyzing more complex processes, the analytical solution in order to obtain an unambiguous relationship can be problematic. In this case, it is advisable to use iterative methods such as the Euler-Cromer algorithm [12], the Runge-Kutta method [20], the Monte Carlo method [9], etc. The accuracy of such models is determined by the step of changing the values of the input parameters that determine the state of the model at a given time.

In the case of interaction of input parameters with each other, the study of the model's functioning nature can be implemented using both Bayesian networks [16, 18] or Petri networks [22]. Therefore, taking into account the presented herein-before, we can conclude that the acquisition of experience regarding creation of models and optimizing their operation mode is an important stage in the preparation of bachelors, masters and doctors of science. In this paper, we present one of the way for creation and investigating the modes of a heating system operating based on the Joule-Thomson effect, the working fluid of which is a real gas.

2 Literature Review

A lot of scientific works are devoted to application of computer simulation techniques in various areas of scientific research [1, 4, 8, 17]. So, in [15], the authors studied the influence of site parameters to the variation of the amplification factor, which is named Fourier Amplification Factor and is defined as the ratio of the Fourier transform of the seismic motion at surface and at bedrock. The proposed model was based on the wave propagation theory and was used limited to 1D linear viscoelastic domain. The new formula explained the investigated dependences has been proposed as the simulation result. In [15], the performance of fish-bone wells was investigated. As a result of the simulation, derived a well-bore and reservoir flow coupling model for fish-bone multilateral wells in the bottom water reservoirs. The created model considered plenty of parameters that may have significant impacts on productivity and pressure drop in the well, including the fish-bone structure, the main and branch well-bores' length, the spacing distance of the branch well-bores, well-bore radius, and preformation parameters. The authors have shown that in comparison with other similar models, the proposed coupling model is more consistent with the results of actual field situation. The papers [2, 3, 5, 10] present the results of the research focused on complex use of data mining and machine learning techniques to gene expression data processing in order to extract the most informative genes in terms of resolution ability of the model to identify the state of the investigated biological objects (health or ill). The authors have proposed the model based on step-wise procedure of the use of clustering and classification techniques. The paper [21] introduce the model based on the potential energy underlying single cell gradients that learns

an underlying differentiation landscape from time-series scRNA-seq data. The authors have shown how the proposed model can simulate trajectories for perturbed cells, recovering the expected effects of known modulators of cell fate in hematopoiesis and pancreatic β cell differentiation. The questions regarding application of Joule – Thomson effect for various processes simulation are considered in [6, 13, 14, 21]. So, in [21], the authors proposed a throttling and heat exchange structure, where heat transfer and throttling coexisted. They fabricated a laminated microchannel distributed Joule – Thomson cooler with pillars utilizing the processing technology of printed circuit heat exchanger. The refrigeration performance concerning the cold-end temperature, cooling power, and temperature distribution has been investigated during the simulation process. In [13], the authors have used six cubic equations of state in order to predict the Joule-Thomson coefficient, specific heat capacity, inversion curve and outlet temperature of the Joule-Thomson valve parameters. The accuracy of each of the equations was estimated by the comparison of experimental data with obtained results. The simulation results have shown that most of the equations showed reasonable prediction on the low-temperature branch of Joule-Thomson inversion curves, but only original Soave-Redlich-Kwong and Patel and Teja equations estimated well at the high-temperature branch. The computational fluid dynamics modelling approach based on the Joule-Thomson effect on gas dehydration and the natural gas liquid was proposed in [14]. The authors analyzed the droplet behavior inside the separator by particle tracing and moisture diffusion methods. In [6], the authors present the research results concerning the design of an apparatus and built for determining the Joule-Thomson effect. The accuracy of the device was verified by comparing the experimental data with the literature on nitrogen and carbon dioxide.

The analysis of the literature review indicates the high actuality of the use of computer simulation methods for modelling various nature processes.

The goal of this paper is the investigation of the modes of a heating system operating, the working fluid of which is a real gas, based on the use of the Joule-Thomson effect.

3 Materials and Methods

3.1 Theoretical Describing the Joule-Thomson Effect

According to the Joule-Thomson effect, when the gas adiabatic expands, its temperature changes [11]. This effect is illustrated in Fig. 1.

Let's mentally select the volume V_1 to the left of the throttle, which occupies the space ABNM. After passing through the throttle the selected portion of gas will occupy the position $M'N'B'A'$ with volume V_2 . The boundary AB moves to the NM position. The work $p_1 \cdot S \cdot AM = p_1 V_1$ (S is the cross-sectional area of the tube) is performed on the gas. The boundary $M'N'$ moves to the position $B'A'$ and, the gas performs the work $p_2 \cdot S \cdot M'A' = p_2 V_2$. The total work done

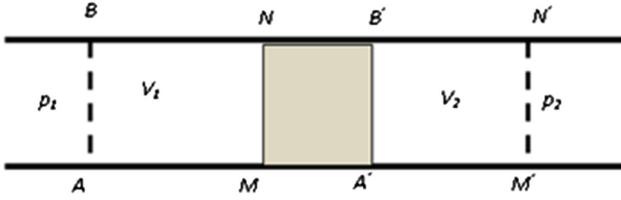


Fig. 1. Adiabatic expansion of the gas: illustration of the Joule-Thomson effect

by the gas is equal to $A' = p_2V_2 - p_1V_1$. In accordance with the first law of thermodynamics for the adiabatic expansion:

$$U_2 - U_1 = A' = -(p_2V_2 - p_1V_1) \quad (1)$$

and

$$U_2 + p_2V_2 = U_1 + p_1V_1 \quad (2)$$

As it can be seen from (2), when the Joule-Thomson effect is performed then the gas enthalpy ($I = U + PV$) does not change. The Van-der-Waals equation can be written as follows:

$$\left(p + \frac{a}{V_M^2}\right)(V_M - b) = RT \quad (3)$$

where p is the gas pressure, V_M is the molar volume of gas, T is the gas temperature, R is the universal gas constant, a and b are corrections depending on the type of gas.

For two states of the gas the Eq. (3) take the forms:

$$p_1V_1 = RT_1 - \frac{a}{V_1} + \frac{ab}{V_1^2} + p_1b \quad (4)$$

$$p_2V_2 = RT_2 - \frac{a}{V_2} + \frac{ab}{V_2^2} + p_2b \quad (5)$$

Then, the gas enthalpy can be expressed as follows:

$$I_1 = U_1 + p_1V_1 = C_pT_1 - \frac{2a}{V_1} + \frac{ab}{V_1^2} + p_1b \quad (6)$$

$$I_2 = U_2 + p_2V_2 = C_pT_2 - \frac{2a}{V_2} + \frac{ab}{V_2^2} + p_2b \quad (7)$$

Taking into account the equality of the enthalpies:

$$C_pT_2 - \frac{2a}{V_2} + \frac{ab}{V_2^2} + p_2b = C_pT_1 - \frac{2a}{V_1} + \frac{ab}{V_1^2} + p_1b \quad (8)$$

we can express the temperatures difference:

$$T_2 - T_1 = \frac{\frac{2a}{V_2} - \frac{2a}{V_1} + p_1b - p_2b + \frac{ab}{V_1^2} - \frac{ab}{V_2^2}}{C_p} \tag{9}$$

Assuming that the initial volume is much smaller than the final one, and the gas pressure at the end of the experiment is insignificant compared to the initial gas pressure, then we can neglect the terms: $\frac{2a}{V_2}$, p_2b , $\frac{ab}{V_2^2}$, and formula (9) takes the form:

$$T_2 - T_1 = \frac{-\frac{2a}{V_1} + p_1b + \frac{ab}{V_1^2}}{C_p} \tag{10}$$

If to express p_1 from (5) and to substitute in (10), we receive:

$$T_2 - T_1 = \frac{\frac{bRT_1}{V_1 - b} - \frac{2a}{V_1}}{C_p} \tag{11}$$

From the formula (11) it can be seen that the temperature increase is determined by the corrections of a and b , i.e. by type of the used gas.

3.2 Calculation of Heating System Efficiency Based on the Joule-Thomson Effect

Let's consider a model of a heating system whose working fluid is gas. In the proposed model, gas is burned in the boiler, due to which energy is released, part of which goes to heat the working fluid (gas). The application of the Joule-Thomson effect implies the presence of two zones in the heating system, and the volume of the first zone should be much smaller than the volume of the second one. As soon as the heated gas has accumulated in the first zone, the pressure in this zone increases. When the value of the pressure difference between the zones reaches a certain boundary value, the valve, which passes the gas into the second zone, opens. As a result of a sharp expansion and reality of the gas, there will be an increase in its temperature. Then most of this gas is pumped out of the pipes and heated again. The cycle repeats.

Between the room and the environment through the walls, the heat exchange is carried out. As a result, an outflow of heat from the room is performed. Concurrently, there is the heat exchange between the room and the pipes through which the heated gas circulates. Due to these two processes, thermal equilibrium is established in the room. Let's calculate the efficiency of this process.

The boiler burns a gas, releasing heat energy, part of which is used to keep in the room a constant temperature. This energy is transmitted to the working fluid, which leads to its heating:

$$\eta_1 Q_1 = c_{gas} m_{gas} (T_1 - T_0) \tag{12}$$

where: Q_1 is the full energy from the gas combustion; η_1 is the boiler efficiency; $c_{\{gas\}}$ is the specific heat of gas; T_0 and T_1 are the initial gas temperature and its temperature after heating respectively; $m_{\{gas\}}$ is the mass of the heated gas.

Then, the gas temperature after heating can be expressed as follows:

$$T_1 = T_0 + \frac{\eta_1 Q_1}{c_{gas} m_{gas}} \quad (13)$$

The Fourier equation for the heat balance between the radiator and the room looks like follows:

$$Q_2 = -\chi_{radiator} \frac{S_{radiator} \tau (T - T_2)}{x_{radiator}} \quad (14)$$

where: $\chi_{radiator}$ is the thermal conductivity coefficient (determined by the material from which the radiator is made); $S_{radiator}$ is the area of the radiator contact with the environment; τ is the time during which the heat exchange took place; T is the room temperature; T_2 is the temperature of the working fluid in the radiator; $x_{radiator}$ is the wall thickness of the radiator.

This heat goes to compensate for heat consumption through the walls. So, on the other hand, Q_2 is equal to:

$$Q_2 = \chi_{walls} \frac{S_{walls} (T - T_3) \tau}{x_{walls}} \quad (15)$$

where: χ_{walls} is the thermal conductivity coefficient of walls; S_{walls} is the area of the walls; T is the room temperature; T_3 is the temperature of the environment; x_{walls} is the wall thickness.

For convenience, enter the notation: $k_{\{radiator\}} = (\chi_{radiator} S_{\{radiator\}}) / x_{\{radiator\}}$, $k_{\{walls\}} = (\chi_{walls} S_{\{walls\}}) / x_{\{walls\}}$. For getting the T_2 , equate the (14) and (15) taking into account the entered notations. As a result, we have:

$$k_{radiator} (T_2 - T) \tau = k_{walls} (T - T_3) \tau \quad (16)$$

and

$$T_2 = \frac{k_{walls}}{k_{radiator}} (T - T_3) + T \quad (17)$$

In accordance with (9):

$$T_1 = \left(\frac{bR}{C_p(V-b)} \right)^{-1} \left(T_2 + \frac{2a}{VC_p} \right) \quad (18)$$

In this case, in accordance with formula (12) to heat the working fluid will require the following amount of heat:

$$Q_1 = \frac{c_{gas} m_{gas}}{\eta_1} \left(\left(\frac{bR}{C_p(V-b)} \right)^{-1} \left(\frac{k_{walls}}{k_{radiator}} (T - T_3) + T + \frac{2a}{VC_p} \right) - T_0 \right) \quad (19)$$

Taking into account that the mass of the used gas is equal to the product of the mass of gas passing through the pipes per unit time and the time of the gas passage:

$$m_{gas} = \frac{\Delta m_{gas}}{\Delta t} \tau \quad (20)$$

we can rewrite the (19) in the following way:

$$Q_1 = \frac{c_{gas} \frac{\Delta m_{gas}}{\Delta t} \tau}{\eta_1} \left(\left(\frac{bR}{C_p(V-b)} \right)^{-1} \left(\frac{k_{walls}}{k_{radiator}} (T - T_3) + T + \frac{2a}{VC_p} \right) - T_0 \right) \quad (21)$$

At the same time, part of the energy expended on the work moving the working fluid through the pipes. This part of the energy can be determined as follows:

$$Q = P\tau \quad (22)$$

where: P is the power of the pump, τ is the time of its operation.

It is obvious, the value of the pump power should be a function of the feed rate of the working fluid through the pipes:

$$P = f\left(\frac{\Delta m_{gas}}{\Delta t}\right) \quad (23)$$

To evaluate, let's consider the power dependence of the pump power on the gas flow rate through the pipes:

$$P = a + \alpha \left(\frac{\Delta m_{gas}}{\Delta t} \right)^n = P_0 + \alpha \left(\frac{\Delta m_{gas}}{\Delta t} \right)^2 \quad (24)$$

where: P_0 is the initial power of the pump; α is the proportionality coefficient.

From formulas (21)–(24) it follows that the total amount of the used heat can be determined in the following way:

$$Q_1^{total} = \frac{c_{gas} \frac{\Delta m_{gas}}{\Delta t} \tau}{\eta_1} \left(\left(\frac{bR}{C_p(V-b)} + 1 \right)^{-1} \left(\frac{k_{walls}}{k_{radiator}} (T - T_3) + T + \frac{2a}{VC_p} \right) - T_0 \right) + (P_0 + \alpha \left(\frac{\Delta m_{gas}}{\Delta t} \right)^2) \tau \quad (25)$$

In this process, the useful energy is determined by the formula (15). The efficiency of the cycle is equal to the ratio of useful and spend energy and is determined by the formula:

$$\eta = \eta_1 k_{walls} (T - T_3) \times \left(c_{gas} \frac{\Delta m_{gas}}{\Delta t} \left(\left(\frac{bR}{C_p(V-b)} + 1 \right)^{-1} \right) \times \frac{k_{walls}}{k_{radiator}} (T - T_3) + T + \frac{2a}{VC_p} \right) - T_0 + P_0 + \alpha \left(\frac{\Delta m_{gas}}{\Delta t} \right)^2 \right)^{-1} \quad (26)$$

3.3 Calculation of Heating System Efficiency Whose Working Fluid is Water

As in the first case, when burning gas in the boiler heat is released, part of which goes to maintain a constant temperature in the pipe:

$$\eta_1 Q_1 = c_{water} m_{water} (T_1 - T_0) \quad (27)$$

When the working fluid of the heating system is water, the Joule-Thomson effect is not observed. Therefore, T_1 is equal to T_2 and is determined by formula (17). Let's substitute this value in formula (27) and express Q_1 :

$$Q_1 = \frac{c_{water} \frac{\Delta m_{water} \tau}{\Delta t} \left(\frac{k_{walls}}{k_{radiator}} (T - T_3) + T - T_0 \right)}{\eta_1} \quad (28)$$

Considering (22), we obtain:

$$Q_1^{total} = \frac{c_{water} \frac{\Delta m_{water} \tau}{\Delta t} \left(\frac{k_{walls}}{k_{radiator}} (T - T_3) + T - T_0 \right)}{\eta_1} + P\tau \quad (29)$$

Taking into account that the useful energy is determined by the formula (15) we receive a formula for determining the of the heat system efficiency:

$$\eta = \frac{\eta_1 k_{walls} (T - T_3)}{c_{water} \frac{\Delta m_{water}}{\Delta t} \left(\frac{k_{walls}}{k_{radiator}} (T - T_3) + T - T_0 \right) + P} \quad (30)$$

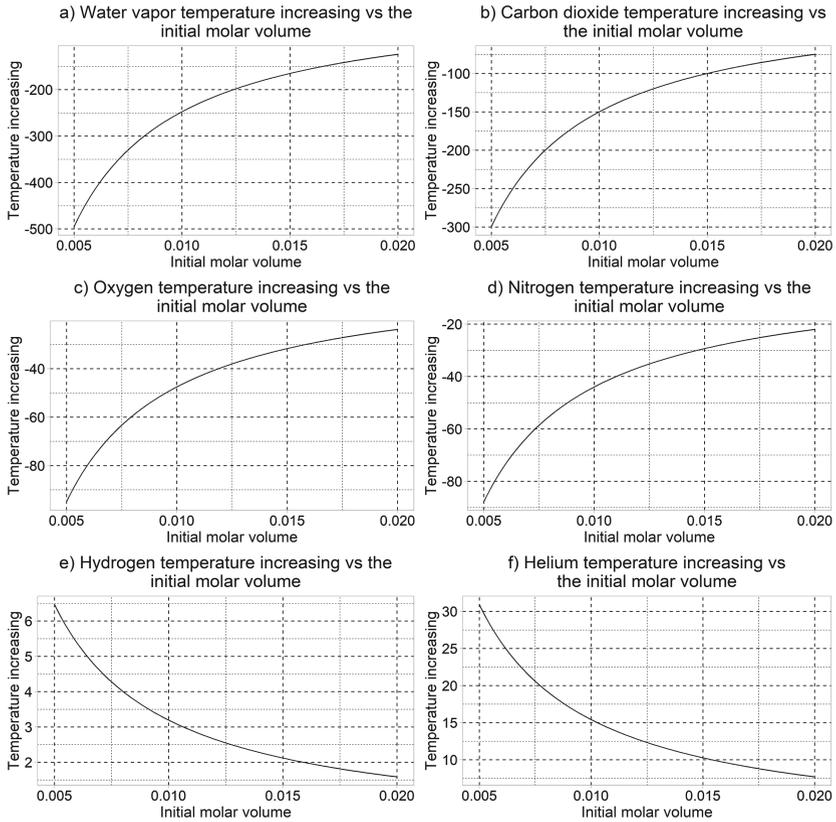
4 Simulation, Results and Discussion

In accordance with (11), increasing the temperature due to the Joule-Thomson effect depends on the Van der Waals amendments a and b. To design the installation, it is necessary to take this into account and choose the gas that will provide the greatest temperature difference. The simulation procedure was performed in the R software [7] using tools of ggplot2 package [19]. The following gases were studied: water vapor, carbon dioxide, oxygen, nitrogen, hydrogen and helium. As a result of the simulation, at the first stage, the charts of the temperature increase owing to its expansion versus the initial volume of the appropriate gas were created. At the second stage, we investigated the efficiency of the heat system created based on the use of both the real gas and water. Table 1 presents the gases parameters which were used during the simulation procedure implementation. Figure 2 shows the simulation results regarding the investigation of the temperature increasing versus the initial molar volume for various types of real gases.

As it can be seen from Fig. 2, in the cases of the use of water vapor, carbon dioxide, oxygen and nitrogen the temperature is decreased. This fact indicates

Table 1. Parameters that were used during the simulation procedure implementation

Type of gas	$a, \frac{Pa \cdot m^6}{mol^2}$	$b, \cdot 10^{-5} \frac{m^3}{mol}$	$R, \frac{J}{mol \cdot K}$	T_1, K
water vapor	0.556	3.06	8.314	273
carbon dioxide	0.364	4.26		
oxygen	0.136	3.16		
nitrogen	0.137	3.9		
hydrogen	0.024	2.7		
helium	0.00343	2.34		

**Fig. 2.** The simulation results concerning investigation of the heat system temperature increasing versus the gas initial molar volume for: a) water vapor; b) carbon dioxide; c) oxygen; d) nitrogen; e) hydrogen; f) helium

that these gases reasonable to use in refrigeration systems. In the case of the use of hydrogen and helium the temperature of the heat system increases. This fact indicates that these gases can be used in the heat systems. Moreover, the comparison of the charts showed in the Figs. 2(e) and 2(d) allows concluding about more effectiveness of helium in terms of energy point of view. However, in terms of security, the hydrogen is more reasonable.

Figure 3 shows the simulation results concerning dependence of the heat system efficiency versus the feed speed of the working substance for the classical model (water is used) and the model based on the use of real gas (hydrogen and helium). Within the framework of the simulation procedure performing, we have used the parameters of a standard one-room flat: efficiency of a steam boiler 91.7%; the thermal conductivity of brick walls 0.5 m thick and with a total area of 200 m^2 $104 \text{ J/s} \cdot \text{K}$; room temperature 290 K; environment temperature 265 K; the thermal conductivity of the iron radiator, wall thickness 2.5 cm and area 1.9 m^2 $3 \cdot 14250 \text{ J/s} \cdot \text{K}$; the initial temperature of gas 290 K; initial power of the pump 5000 Wt; the coefficient of proportionality between the pump power and the feed rate of the working fluid through the pipes 5000 J/kg.

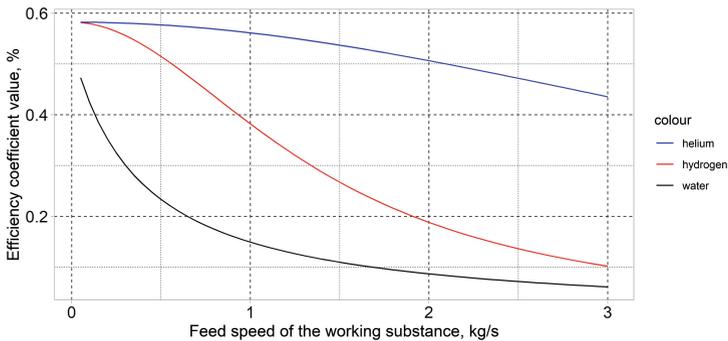


Fig. 3. The charts of the heat system efficiency versus the feed speed of the working substance for the classical model (water is used) and the model based on the use of real gas (hydrogen and helium)

The obtained results confirm the assumption about higher efficiency of the heating system which is based on the Joule-Thomson effect in comparison with the standard heating system that is based on water (the working fluid). Moreover, the simulation results have shown also significantly higher efficiency of the heating system based on helium in comparison with the system were used hydrogen as the working gas. However, in terms of security, the use of hydrogen as the working substance is more reasonable.

Figure 4 presents the charts of the ratio of the heating systems efficiency, the working fluid of which is real gas (hydrogen and helium) to the standard heating system where the working fluid is water.

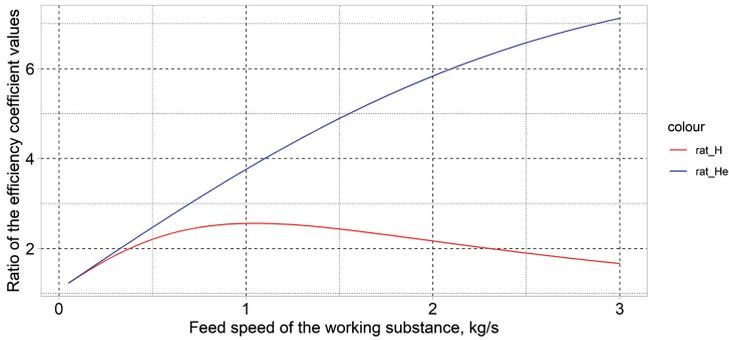


Fig. 4. The charts of the ratio of heat systems efficiency coefficients versus the feed speed of the working fluids

An analysis of Fig. 4 allows concluding that in the instance of the use of hydrogen as the heating system working fluid, it is possible to determine the optimal conditions of the system application in terms of the maximum value of the system efficiency in comparison with the standard heating system. This condition corresponds to the extreme value of the function showed in Fig. 4 in the red color.

5 Conclusions

An application of computer simulation techniques for the analysis of various processes allows us to understand both nature and particularities of the appropriate process in order to optimize the parameters of the system which is based on this process carried out. The skills which students obtain within the course focused on computer simulation of physical processes may allow them to successfully find a job both in IT companies and in small enterprises for various purposes.

In this paper, we have presented the model of a heating system based on the Joule-Thomson effect where the real gas was used as the working fluid. The following gases have been studied within the framework of the research: water vapor, carbon dioxide, oxygen, nitrogen, hydrogen and helium. As a result of the simulation, at the first stage, the charts of the temperature increase owing to its expansion versus the initial volume of the appropriate gas have been created. At the second stage, we have investigated the efficiency of the heat system created based on the use of both the real gas and water. The analysis of the simulation results has shown that in the cases of the use of water vapor, carbon dioxide, oxygen and nitrogen the temperature is decreased. This fact indicates that these gases reasonable to use in refrigeration systems. In the case of the use of hydrogen and helium the temperature of the heat system increases. This fact indicates that these gases can be used in the heat systems. Moreover, the comparison of the obtained charts allows concluding also about more effectiveness of helium in terms of energy point of view. However, in terms of security, the hydrogen is more reasonable.

References

1. Andrunyk, V., Vasevych, A., Chyrun, L., et al.: Development of information system for aggregation and ranking of news taking into account the user needs. In: CEUR Workshop Proceedings, vol. 2604, pp. 1127–1171 (2020)
2. Babichev, S., Škvor, J.: Technique of gene expression profiles extraction based on the complex use of clustering and classification methods. *Diagnostics* **10**(8), 584 (2020). <https://doi.org/10.3390/diagnostics10080584>
3. Babichev, S., Lytvynenko, V., Škvor, J., et al.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction. In: Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, p. 8478452 (2018). <https://doi.org/10.1109/DSMP.2018.8478452>
4. Babichev, S., Osypenko, V., Lytvynenko, V., et al.: Comparison analysis of biclustering algorithms with the use of artificial data and gene expression profiles. In: 2018 IEEE 38th International Conference on Electronics and Nanotechnology, ELNANO 2018, p. 8477439 (2018). <https://doi.org/10.1109/ELNANO.2018.8477439>
5. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopolym. Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
6. Gao, M., Wang, L., Chen, X., et al.: Joule-Thomson effect on a CCS-relevant ($\text{CO}_2 + \text{N}_2$) system. *ACS Omega* **6**(14), 9857–9867 (2021). <https://doi.org/10.1021/acsomega.1c00554>
7. Ihaka, R., Gentleman, R.: R: a language for data analysis and graphics. *J. Comput. Graph. Stat.* **5**(3), 299–314 (1996)
8. Izonin, I., Tkachenko, R., Verhun, V., et al.: An approach towards missing data management using improved GRNN-SGTM ensemble method. *Int. J. Eng. Sci. Technol.* **24**(3), 749–759 (2021). <https://doi.org/10.1016/j.jestch.2020.10.005>
9. Kang, D., Shin, Y., Jeong, S., et al.: Development of clinical application program for radiotherapy induced cancer risk calculation using Monte Carlo engine in volumetric-modulated arc therapy. *Radiat. Oncol.* **16**(1), 108 (2021). <https://doi.org/10.1186/s13014-020-01722-0>
10. Lytvynenko, V., Lurie, I., Krejci, J., Voronenko, M., Savina, N., Taif, M.A.: Two step density-based object-inductive clustering algorithm. In: CEUR Workshop Proceedings. vol. 2386, pp. 117–135 (2019)
11. Roy, B.: *Fundamentals of Classical and Statistical Thermodynamics*. John Wiley & Sons, New York (2002)
12. Saroja, G., Nuriyah, L.: Numerical solution of nonlinear vibration by Euler-Cromer method. *IOP Conf. Ser. Mater. Sci. Eng.* **546**(3), 032029 (2019). <https://doi.org/10.1088/1757-899X/546/3/032029>
13. Shoghl, S., Naderifar, A., Farhadi, F., Pazuki, G.: Comparing the predictive ability of two- and three-parameter cubic equations of state in calculating specific heat capacity, Joule - Thomson coefficient, inversion curve and outlet temperature from Joule - Thomson valve. *Cryogenics* **116**, 103288 (2021). <https://doi.org/10.1016/j.cryogenics.2021.103288>
14. Shoghl, S., Naderifar, A., Farhadi, F., Pazuki, G.: Optimization of separator internals design using CFD modeling in the Joule-Thomson process. *J. Nat. Gas Sci. Eng.* **89**, 103889 (2021). <https://doi.org/10.1016/j.jngse.2021.103889>

15. Stambouli, A., Zendagui, D., Bard, P., Dif, H.: Influence of site parameters on Fourier amplification application for 1D linear viscoelastic method. *Periodica Polytechnica Civ. Eng.* **65**(1), 229–241 (2021). <https://doi.org/10.3311/PPci.12601>
16. Sun, B., Zhou, Y., Wang, J., Zhang, W.: A new PC-PSO algorithm for Bayesian network structure learning with structure priors. *Expert Syst. Appl.* **184**, 115237 (2021)
17. Tkachenko, R., Izonin, I., Kryvinska, N., et al.: An approach towards increasing prediction accuracy for the recovery of missing IoT data based on the GRNN-SGTM ensemble. *Sensors (Switzerland)* **20**(9), 2625 (2020). <https://doi.org/10.1016/j.jestch.2020.10.005>
18. Turkia, J., Mehtätalo, L., Schwab, U., Hautamäki, W.: Mixed-effect Bayesian network reveals personal effects of nutrition. *Sci. Rep.* **11**(1), 12016 (2021). <https://doi.org/10.1038/s41598-021-91437-3>
19. Wickham, H.: *ggplot2: Elegant Graphics for Data Analysis*. Springer, New York (2016). <https://doi.org/10.1007/978-3-319-24277-4>
20. Xin, X., Qin, W., Ding, X.: Continuous stage stochastic Runge–Kutta methods. *Adv. Differ. Eqn.* **2021**(1), 1–22 (2021). <https://doi.org/10.1186/s13662-021-03221-2>
21. Yeo, G., Saksena, S., Gifford, D.: Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions. *Nat. Commun.* **12**(1), 3222 (2021). <https://doi.org/10.1038/s41467-021-23518-w>
22. Zeinalnezhad, M., Chofreh, A., Goni, F., Hashemi, L., Klemeš, J.: A hybrid risk analysis model for wind farms using coloured petri nets and interpretive structural modelling. *Energy* **229**, 120696 (2021). <https://doi.org/10.1016/j.energy.2021.120696>



Simulating Soil Organic Carbon Turnover with a Layered Model and Improved Moisture and Temperature Impacts

Olha Stepanchenko¹, Liubov Shostak¹, Viktor Moshynskiy¹,
Olena Kozhushko¹, and Petro Martyniuk¹

National University of Water and Environmental Engineering, Rivne, Ukraine
{o.m.stepanchenko,l.v.shostak,v.s.moshynskiy,o.l.d.kozhushko,
p.m.martyniuk}@nuwm.edu.ua

Abstract. Decomposition of soil organic carbon (SOC) is an important part of the global carbon cycle, and is connected to multiple ecological processes. A comprehensive understanding of the process and subsequent improvement of soil management practices is an important step in developing sustainable agriculture and other land uses. This can not only increase the quality and productivity of arable lands, but also reduce carbon dioxide emissions to the atmosphere.

Currently developed SOC models are characterized by the multitude of considered factors: biological and physical processes, chemical reactions, influence of weather, crop and soil conditions. The importance of subsoil carbon also has been recognized, and a number of layered SOC models emerged. As decomposition of organic matter depends greatly on the soil physical characteristics such as moisture and temperature, these layered carbon models require corresponding estimates to remain consistent.

In this study, we take the RothPC-1 layered carbon decomposition model, and supply it with a comprehensive physical soil water and heat flow model. The two models are interconnected in a way that the carbon model uses the estimates of soil moisture and temperature from the moisture model, and the physical soil properties are then modified according to the simulated content of organic matter.

The model simulations were conducted for three sites in Ukraine. The experiment covered a period between 2010 and 2020 and involved only data from the open datasets. The results demonstrate the behavior of layered soil carbon decomposition model under different climate conditions.

Keywords: Soil organic carbon · Carbon turnover · Mathematical modeling · Soil moisture · CO₂ emission

1 Introduction

Soil organic carbon (SOC) content is among the primary factors contributing to soil quality and fertility. Along with the climate and water productivity, it

is marked as one of the most important factors that define soil health [30,40]. SOC information is important not only for assessing the soil state and planning fertilizer applications, but for increasing awareness on long-lasting impacts of human activities, ensure ecological and food security. Intensive exploitation of agricultural lands caused depletion of soil productivity potential. Thus, understanding SOC turnover is crucial for developing sustainable agriculture, forestry and other land use strategies, and calls for constant monitoring and control of the quantity and quality of natural resources [15,37].

Like many other ecosystem processes, SOC turnover is heavily influenced by atmosphere conditions, and also produces a feedback in the form of CO₂ respiration. Soil respiration is one of the major components of the carbon cycle. A significant role in the process is played by the microorganisms, whose activity is influenced by soil temperature and moisture. The excess of emitted CO₂ due to human activities is causing the greenhouse gas effect. As the global temperatures rise due to global climate change, some concerns have been raised as to the possible increase of global CO₂ emission to the atmosphere [23].

Over the last decades, numerous models have been developed to simulate and predict the process of carbon decomposition [3,5,7,20,26,32,35,41]. However, most of them are designed to simulate the organic matter turnover in the top 20-30 cm of the soil, whereas a significant amount of organic compounds is found along one meter of the soil depth. Taking into account these deeper soil layers allows to estimate soil carbon stocks more comprehensively, and also leads to the concept of interaction between soil layers.

Recent studies also tend to couple the carbon models with atmospheric and biophysical models. Most common is cooperation of SOC models with plant growth simulators. CENTURY was initially combined with a biomass growth model [27], and later evolved into DayCent which also tracks nutrient uptake and gas emission from the soil. It has also been coupled with a commercial crop growth model DSSAT [39]. Another crop simulation model, WOFOST, included various nutrient mineralization submodels in its implementations, e.g. [33]. DNDC model includes a one-dimensional water flow model to account for moisture effects and nutrient leaching [24]. A rigorous consideration of related physical and biophysical processes should further become the goal for developing and refining carbon turnover models.

The main contributions of this paper are summarized as follows: 1. The layered SOC decomposition model is coupled with a detailed soil moisture and temperature model. 2. The model is complemented with abiotic stress functions that more accurately account for the subsoil state. 3. The numerical experiment is conducted to demonstrate the impact of the layered model and atmospheric conditions on the SOC stocks, composition and CO₂ emissions. The rest of the paper is structured in the following way. Section 2 briefly reviews current development of modeling SOC decomposition and the principal contributing factors, such as soil composition and moisture. Section 3 describes the mathematical models employed in the developed computer system: RothPC model for carbon

decomposition, physical soil moisture and temperature model, and relations for abiotic impacts to the carbon model. Section 4 describes the data sources and the experimental setting for verifying the model. Section 5 presents the modeling results and discusses the impacts of deeper soil layers in the carbon turnover model. In conclusions, we summarise the differences of predicted SOC dynamics with topsoil and layered models, and describe the directions for possible research to further improve the model consistency and rationale.

2 Literature Review

Decomposition of organic matter has a highly nonlinear nature and is influenced by multiple factors. It comprises a range of biological, chemical and mechanical processes that are not perfectly understood. At the same time, biochemical soil analysis is rather costly, and the measurement data are scarce. Experiments are mostly done in laboratories, isolated from environmental influence, and so the results are often conflicting with field observations and global studies [8].

In the last decades, various organic matter models arose, which reflect different ways of describing and simulating the SOM decomposition processes. Rothamsted Carbon (RothC) model considers only the soil organic carbon (SOC) turnover, which constitutes roughly about 60% of total SOM [7]. The decomposition is modeled as a first-order process, meaning the decomposition rate depends linearly on the current amount of organic matter. The CENTURY model uses a similar approach, and also takes into account nitrogen, phosphorus and sulfur transformation [26]. MESDM describes the decomposition in terms of microbial activity, and concentrates on microbes and their byproducts [41]. The biological processes are described here with so-called Monod kinetics so that decomposition rates depend on the amount of microbial biomass involved in the reaction. CWM1, a model designed specifically for constructed wetlands, does not distinguish between different SOM components, but considers different types of bacteria, as well as inorganic nitrogen and sulfur compounds produced by SOM degradation process [22].

The representation of organic carbon composition in the models also differs. RothC considers input plant litter (divided into readily decomposable and resistant material), humus, microbial biomass, and inert organic matter. CENTURY also separates litter compartments and the active SOC pool (including microbes) into near-surface and underground soil layers. A more recent model adaptation with daily time step, DayCent, considers similar pools for SOC as RothC with addition of nitrogen compounds and gases [9]. MESDM differentiates between soil carbon, dissolved organic carbon, microbial biomass and extracellular enzyme. CWM1 covers complex chemical processes in 16 organic and mineral soil compartments, including nitrogen and sulfur compounds. For a detailed review of these and other SOC models, we refer to Shibu et al. [32].

SOC stocks and turnover rates are heavily influenced by multiple environmental factors. Study [16] indicates that SOC stock data at 5 cm had the most significant correlation with soil moisture (0.58 Pearson correlation), sand content (-0.57) and soil temperature (-0.46). Most models introduce rate modifiers for soil moisture, temperature and texture, and some models also consider crop cover, air temperature, pH and other factors [24]. These are often called abiotic stress or limiting factors. Examples of abiotic stress functions may be found in [7, 17, 20, 26, 35] and in the review [32].

The influence of soil temperature might be the most controversial of these factors. Numerous laboratory studies indicate it varies for different carbon pools, and is inversely proportional to SOC decomposability [43]. However, global studies mostly find no such relationship [8]. Other studies point out the connections between temperature impact and soil texture. Later studies tend to detract the importance of soil temperature in SOC decomposition [13, 17]. More so, the effects of global warming on soil carbon emissions are estimated to be not significant in recent studies [10, 25]. Another argument for SOC sustainability during global warming was found in SOC downward movement. As the fraction of incoming organic material from the environment is transferred to deeper soil layers, it is ‘conserved’ in a way from the influence of atmospheric temperature. This moving fraction, moreover, is subtracted from CO₂ emission, and the CO₂ released in the lower soil layers is not immediately lost to the atmosphere, but can be partly assimilated back into soil carbon.

Considering deeper soil layers thus results in lower (and more realistic) simulated CO₂ emission. For this reason, some SOC models have been extended to account for subsoil carbon and its downward movement. CENTURY implemented a multi-layered SOC transformation version [12]. The vertical distribution of carbon is modeled here with an exponential curve based on the exponential filter. Layered version of RothC-26.3, called RothPC-1, divides the soil into 9-inch layers. On each layer, the same decomposition processes are transpiring, but some fraction of carbon from each of the pools ‘falls’ down to the underlying layer. As this new ‘sink’ term is introduced, another model sink – CO₂ emission – decreases. The model succeeded in describing subsoil SOC dynamics and CO₂ emission more accurately. However, like the original RothC-26.3 model, it uses average monthly air temperature and soil moisture deficit in the top soil layer for temperature and moisture stress. These values must affect deeper soil layers differently than the soil surface, as the authors of Roth-PC pointed out themselves [14].

Despite the advanced state of carbon turnover modeling, consideration of subsoil SOC dynamics had not yet received sufficient attention. Many studies point out the importance of SOC movement between layers, but only a few comprehensive models exist, and they still require further development. Taking into account the influence of soil temperature and moisture in the soil layers is one of the problems that need more thorough consideration. In this study, we attempt to tackle this problem for the RothPC-1 model. We complement the carbon

turnover model with a physical soil model for belowground moisture and temperature, and use more applicable relations to access their impact. The results are aimed to demonstrate the impact of accounting downward SOC movement in the model and accurate consideration of abiotic effects on carbon decomposition.

3 Mathematical Model

3.1 Layered SOC Decomposition Model

The layered SOC decomposition is well described in the RothPC-1 model. It is based on the earlier model RothC-26.3 for topsoils, which is grounded on 100 years of soil studies. The model is widely employed to simulate SOC dynamics, notably, in the framework of Global Soil Organic Carbon Sequestration Potential Map project by Global Soil Partnership (GSP) and Food and Agriculture Organization of the United Nations (FAO) [28], and in creating of SOC maps of China [38].

The structure of pools and flows in the RothC model is shown on Fig. 1. First, incoming plant material is split between decomposable plant material (DPM) and resistant plant material (RPM) pools. For instance, for most agricultural crops the default DPM/RPM ratio is 1.44, and goes as low as 0.25 for woodlands. Then all SOC pools undergo a decomposition process in the same pattern. Decomposed organic carbon is partitioned between humus (HUM), microbial biomass (BIO) pools and CO₂ that leaves the system. The fraction of formed CO₂ is controlled by clay content, and the rest is split in a constant ratio: 54% to HUM and 46% to BIO. Decomposition rates are determined by pool-specific constants, and multiplied by factors for temperature, moisture and soil cover effects. Thus, HUM and BIO are decomposed in a feedback pathway, and DPM and RPM are supplied only with the incoming plant material. Inert organic matter (IOM) pool takes no part in decomposition and remains constant throughout the simulation.

The process can be represented with the following set of first-order differential equations:

$$\begin{aligned} \frac{dC_i}{dt} &= K_i A C_i + F_{litter,i} L, i \in \{DPM, RPM\}; \\ \frac{dC_i}{dt} &= K_i A C_i + \sum_{j=1}^4 K_j A C_j (1 - F_{CO_2}) F_i, i \in \{BIO, HUM\}; \\ \frac{dC_i}{dt} &= 0, i \in \{IOM\}.. \end{aligned} \quad (1)$$

Here, C_i is the amount of SOC in the i th pool, K_i is the constant base decomposition rate (inverse of the pool's decomposition time), $F_{litter,i}$ is the split factor for incoming plant material between DPM and RPM, F_i is the constant split factor between BIO and HUM during decomposition, F_{CO_2} is the fraction of carbon that transforms to CO₂, $A = A_w * A_t * A_c$ is the combined impact of

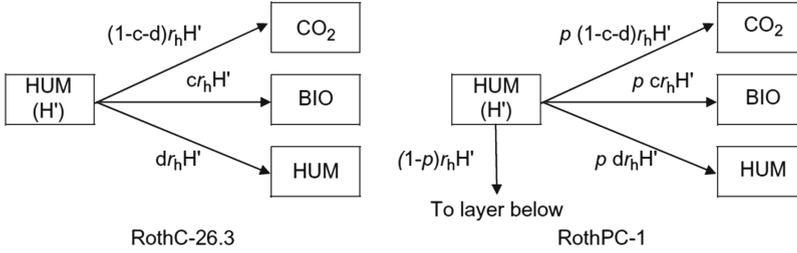


Fig. 2. Comparative schemes of decomposition in RothC-26.3 and RothPC-1 models. The same quantity of organic C is mobilized from each pool (here illustrated by the HUM compartment). Here r_h is the HUM decomposition rate, c is the fraction of decomposed HUM that goes to new BIO, d – the fraction that goes to new HUM. These modifiers are identical for both models. Source: Jenkinson and Coleman, 2008 [14]

For this study, we do not implement an agricultural model to simulate crop growth and senescence, but rather consider a pasture with moderate amounts of litter inputs. We found that the value of 0.8 for parameter p suits this problem setting. We also replace the monthly time step, used in RothC, with a daily one in our realization. This would demonstrate the influence of soil moisture and temperature influence more clearly, especially as the corresponding data are easily available since the physical model, described below, operates on an even finer scale. The set of equations (1) and its corresponding analogues for the layered model together with the initial conditions constitute a Cauchy problem, which we solve with the classic Runge-Kutta method.

3.2 Soil Moisture and Temperature Model

To simulate subsurface water flow in the soil, we use a Richards equation based model. This equation is a fundamental physical equation for both saturated and unsaturated water flow. Richards equation is widely used in comprehensive hydrological simulation models, which often become part of global hydrological and environmental models [6]. HYDRUS, in particular, has been coupled with some agricultural models [31, 42]. It is a very detailed framework suitable for field studies, when most common soil and site parameters are available. In this study, we use a more satellite-based approach, designed for cases when actual field parameters cannot be measured otherwise. It has been tested against in-situ soil measurements in a hydrological problem setting with addition of data assimilation [18, 19], and also has been used together with the CENTURY model to simulate near-surface SOC dynamics [34].

The model consists of coupled one-dimensional nonlinear moisture transport and heat transfer boundary value problems.

$$\frac{\partial \theta(x, t)}{\partial t} = \frac{\partial}{\partial x} \left(k(h) \frac{\partial h}{\partial x} - k_T \frac{\partial T}{\partial x} \right) - \frac{\partial k(h)}{\partial x} - R(h, x, t),$$

$$\left(k(h) \frac{\partial h}{\partial x} - k_T \frac{\partial T}{\partial x} \right) \Big|_{x=0} = Q(t) - E_s(t), t \geq 0, \quad (2)$$

$$\frac{\partial h}{\partial x} \Big|_{x=l} = 0, t \geq 0,$$

$$h(x, 0) = h_0(x), x \in [0; l];$$

$$c_T \frac{\partial \theta(x, t)}{\partial t} = \frac{\partial}{\partial x} \left(\lambda(h) \frac{\partial T}{\partial x} \right) - \rho c_w u(h) \frac{\partial \theta(x, t)}{\partial x},$$

$$T(x, t) \Big|_{x=0} = T_1(t), t \geq 0, \quad (3)$$

$$T(x, t) \Big|_{x=l} = T_2(t), t \geq 0,$$

$$T(x, 0) = T_0(x), x \in [0; l].$$

Here, θ is absolute soil moisture, h is the pressure head, T is the temperature, k is the soil hydraulic conductivity, k_T is the hydraulic conductivity due to temperature, $R(h, x, t)$ is the root water uptake, $Q(t)$ is the precipitation rate, $E_s(t)$ is the soil evaporation rate, $\lambda(h)$ is the soil thermal conductivity, $c_T = c_n(1 - \theta) + c_w\theta$ is the volumetric heat capacity of porous medium, c_n and c_w are the specific heat capacity of solid soil and water, ρ is the soil water density, $u(h)$ is the soil water flux rate. In the boundary conditions, $Q(t)$ is precipitation rate, $E_s(t)$ is soil evaporation rate, $T_1(t)$ and $T_2(t)$ are temperatures on the soil surface and l m depth; $h_0(x)$, $T_0(x)$ are initial conditions for pressure head and temperature, respectively.

The described problem is limited to a one-dimensional case due to its global-scale focus. Possible horizontal diffusion and subsurface flows are neglectable on a regional and even on a field scale. This, however, does not apply to surface water flows, which in this case must be accounted for in the inflow term Q . Among the most notable problems associated with Richards equation is choice of relations between moisture and pressure head, since both are present in the equation. In our model, we use Mualem–van Genuchten model, represented by the following equations:

$$\theta(h) = \theta_{min} + \frac{\theta_{max} - \theta_{min}}{(1 + (-\alpha h)^n)^m}, m = 1 - \frac{1}{n}, \quad (4)$$

$$k(h) = k_s S^l \left(1 - \left(1 - S^{\frac{1}{m}} \right)^m \right)^2,$$

where θ_{min} , θ_{max} are the residual (minimum) and saturation (maximum) water content, k_s is the saturated soil hydraulic conductivity, $S = (\theta - \theta_{min}) / (\theta_{max} - \theta_{min})$ is the saturation degree, α , n and l are the empirical model parameters.

The moisture equation (2) is then transformed using (4) to operate with only pressure heads as unknown variables. Both problems are solved with the

finite difference method on the grid with 5 cm step by depth and 0.1 d by time. Samarskii iterative scheme is employed to deal with nonlinearities in the equations. The solving process and other features of the model, including possible addition of satellite data via data assimilation, are described in detail in [19].

3.3 Abiotic Stress Functions

The described models are connected with the so called abiotic stress functions, describing the effect of moisture and temperature on decomposition processes. Each of the carbon flows in the model is multiplied by the abiotic effect $A = A_t * A_w$, which is thus limiting the volume of the carbon turnover.

Original function for the temperature effect in RothC is presented in equation

$$A_t = \frac{47.91}{1 + e^{\left(\frac{106.06}{T+18.27}\right)}}, \quad (5)$$

where T is the air temperature. In fact, use of air temperature is called ‘a gross oversimplification’ by the authors of RothPC. Another concern over the relation might be raised by its exponential form, when later studies suggest the temperature has a weaker influence.

The moisture effect is described by a piecewise linear function

$$A_w = \begin{cases} 1.0, & \text{if } acc.TSDM < 0.444 \max.TSDM, \\ 0.2 + (1.0 - 0.2) \frac{\max.TSDM - acc.TSDM}{\max.TSDM - 0.444 acc.TSDM}, & \text{otherwise,} \end{cases} \quad (6)$$

where $TSM D$ signifies total soil moisture deficit. Its maximum is defined by the soil clay content, and accumulated value is calculated during simulation with monthly rainfall and precipitation. Basically, it is intended as a simplified estimation of soil moisture in absence of an actual model. To couple RothC with our soil moisture model, we replace this equation with a relation that uses volumetric soil moisture (7).

$$A_w = 1 / (1 + 4 \cdot e^{-6RWC}), \quad (7)$$

Here, RWC is relative soil moisture, defined as in equation (2). This relation is taken from the ANIMO model [21].

4 Experiment

4.1 Experimental Setting

In order to demonstrate the presented model, we conduct numerical experiments for a period between 2000 and 2010. We selected three arbitrary agricultural sites across Ukraine, located in different zones of temperate climate and named after the nearest town. Table 1 presents information on chosen locations, their climatic characteristics and soil composition. Average annual air temperature

Table 1. Experimental sites and their climate characteristics

Site name	Coordinates (latitude, longitude)	Soil organic carbon stock (t/ha)	Soil composition and texture class	Average annual air temperature, °C	Average annual precipitation sum, mm
Rivne oblast, Stepan	51.136962, 26.276769	44	sand – 57% clay – 15% (Sandy Loam)	7.0	610
Zakarpattya, Vynogradiv	48.180321, 22.984819	51	sand – 29% clay – 31% (Clay Loam)	9.7	748
Kherson oblast, Novopavlivka	46.333222, 33.424405	49	sand – 26% clay – 42% (Clay)	9.9	398

and precipitation data are from Weatherbase [2], soil data are downloaded from SoilGrids [29] (see the following section). SOC data are given for the top 30 cm of the soil, and mineral soil composition – for the top 20 cm. Silt content is omitted here since it constitutes the remainder of the mineral soil fraction.

The simulation is then carried out in two frameworks for each site: with topsoil RothC-26.3 model (TM) and layered RothPC-1 model (LM). In both cases, the top soil layer receives the same high amount of plant litter during the growth season, from April to August.

4.2 Data Sources

The developed model is primarily designed to be used worldwide and conduct simulations for any given land site. Thus we generally labor under the assumption that no field measurements are available, and only open datasets can be used for simulations.

Soil texture characteristics (percent of clay and sand) and carbon stock estimations are taken from SoilGrids [29]. The service provides worldwide physical and chemical soil data on a 250 m grid, as estimated from the available soil surveys. Soil texture data and carbon stocks are used to calculate hydrological characteristics after [4]. These parameters are updated on each model step, as the organic carbon content is changing.

The carbon stock values from the database are also used as initial conditions for the SOC model. They are distributed downwards with an exponential filter, according to the concept used in the layered variation of CENTURY model [12]. Then the values are separated into the SOC pools with the following proportions (selected based on the study [1]): 6% to microbial biomass, 75% to humus, 3% to inert organic carbon, 3% and 14% – to decomposable and resistant plant material, respectively. Initial conditions for soil moisture are field capacity, calculated as moisture on -0.33 bar hydraulic head.

Table 2. Datasets and variables used for the numerical experiment

Data category	Source database	Spatial resolution	Temporal resolution	Model parameters
Weather	Reanalysis	ERA 5 Land	0.1° hourly	Skin temperature Total evaporation Surface solar radiation downwards
	LAMETSY	— (nearest meteo station)	daily	Maximum temperature Minimum temperature Average temperature Precipitation
Soil parameters	Soil Grids 2.0	250 m	—	Soil organic carbon stock Clay content Sand content

The initial state of the system, calculated in this way, is often not optimal from the perspective of the model physics. We employ an early start approach to mitigate the instability at simulation start. In the numerical experiments below, the actual calculations start from the year 2005, but the results before 2010 are discarded and excluded from analysis.

Most common weather characteristics for running the simulation are downloaded from the LARge METeorological SYstem (LAMETSY) [36]. It is a dataset of historical data from meteorological stations, as recorded in the NOAA database, provided with an advanced selection API. Other meteorological parameters are assumed from ERA-5 reanalysis database [11]. The full list of parameters and used datasets is presented in Table 2.

5 Results and Discussion

The calculations for the layered model (LM) are carried out at four 20 cm-deep soil layers. The resulting SOC amounts by pools on all layers for a single location (Kherson oblast, Novopavlivka) are presented in Fig. 3. Part of the vertical axis in the plot is cut out to fit the values of the largest pool, i.e. humus.

The visible wave patterns on the plots, especially RPM and DPM, represent the plant material inputs during the season. The SOC stocks in these pools of the topsoil are rapidly decreasing, indicating the local climatic conditions hasten the decomposition rates. This rapid decomposition affects the HUM carbon pool. Since the layered model reallocates a fixed fraction of decomposed carbon to the lower soil layers, a significant amount of SOC is ‘leached’ from the topsoil. The other soil layers, on the contrary, increase their HUM stocks. The increment of HUM on the second layer alone is roughly equivalent to HUM lost in the top layer. Thus, the amount of SOC in the soil is generally increasing.

Figure 4 gives a view on the differences between the LM and TM simulation, as well as between conditions on the chosen sites. The fastest decomposition rates

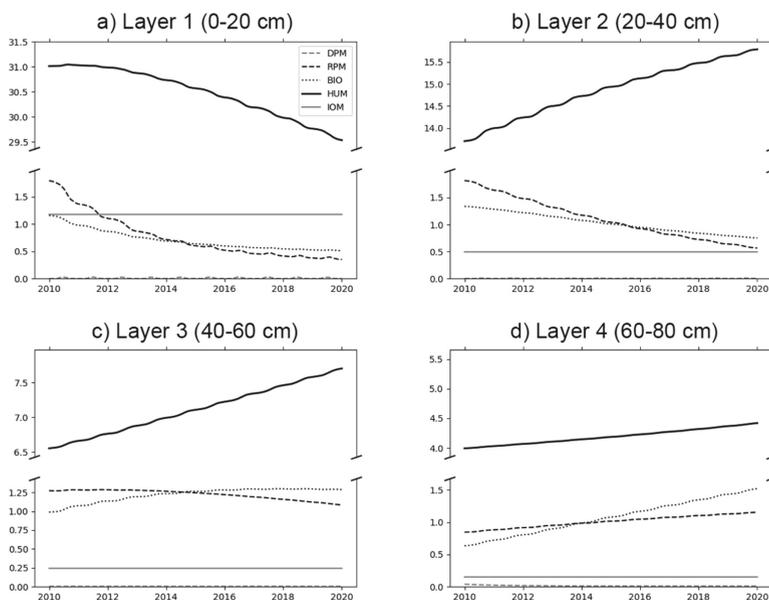


Fig. 3. Simulated soil organic carbon stocks on different soil layers on a site near Kherson, Novopavlivka, in t/ha. Line styles represent SOC pools: resistant organic matter (dashed black line), decomposable OM (dashed gray line), microbial biomass (dotted line), humus (solid black line) and inert OM (solid gray line)

are found on Vynorgadiv and Novopavlivka sites, indicated by rapidly decreasing RPM stocks and high HUM values. The climatic conditions are wet on one site and arid and warm on the other, which speeds up the carbon turnover. High decomposition speed, however, increases the downward movement of SOC to the lower layers. For this reason, the HUM stocks are significantly reduced in LM simulations.

It may also be noted that some carbon pools do not reach a ‘balanced’ point throughout the simulation: RPM stocks in the soil are decreasing on most layers, and BIO values are increasing on the bottom layer. These dynamics are mostly due to imperfect distribution of initial carbon stocks between the layers and pools. We used universal proportions to allocate SOC between the pools on all layers; however, they clearly must have a more complex distribution which must be investigated further.

Stepan site demonstrates the largest simulated values of HUM, increasing even in the layered model. The amplitude of seasonal ‘waves’, as well as changes in RPM during the simulation, are the smallest on this site compared to the others, and major part of the plant material resides in the soil in undecomposed state. The climate in this part of Ukraine is colder, which must be the reason for lower overall decomposition rates.

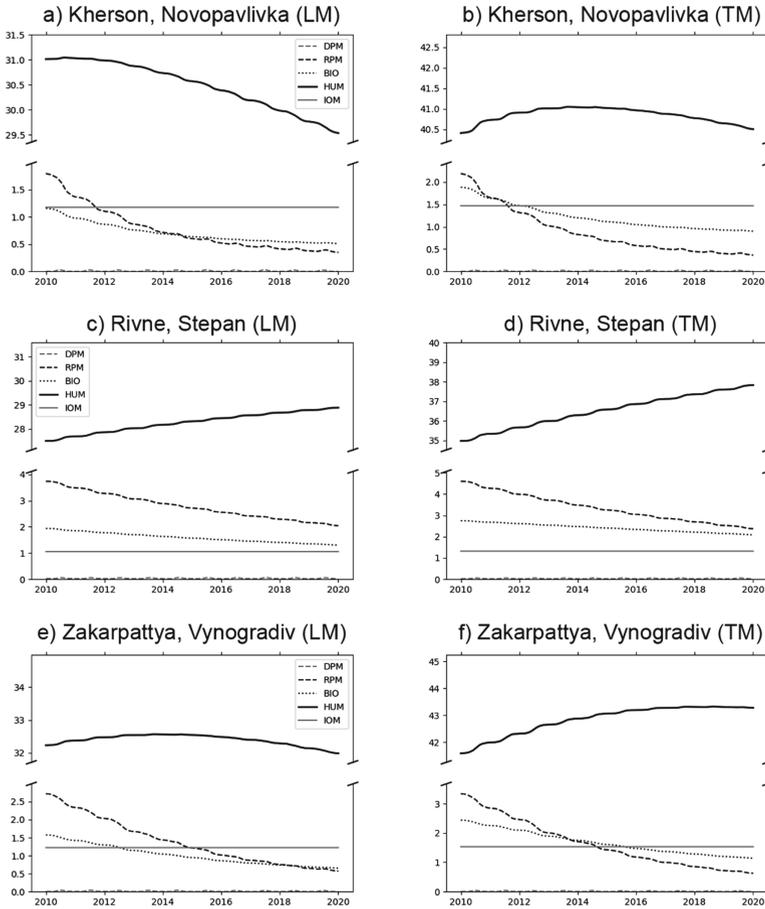


Fig. 4. Simulated soil organic carbon stocks with LM (a, c, e) and TM (b, d, f) model variations by sites, in t/ha. Line styles correspond to SOC pools as in Fig. 3

Generally, carbon stocks in the top soil layer are higher in the TM case, despite the fact identical initial values were set. This difference is obviously due to the downward carbon movement in the layered model. Moreover, leaching to the subsoil also takes its place, as a fraction of SOC leaves the lowest soil layer to the untracked domain. However, the sum of total SOC in the layers is still generally higher than in the topsoil model, and increases with time.

Among the reasons for SOC growth in the layered model is reduced simulated carbon dioxide emission. The monthly sums of produced CO₂ amounts by sites are given in Fig. 5. Here we only count CO₂ formed in the top soil layer to be consistent in comparison, more so as the gas produced on the lower soil layers is not directly emitted to the atmosphere.

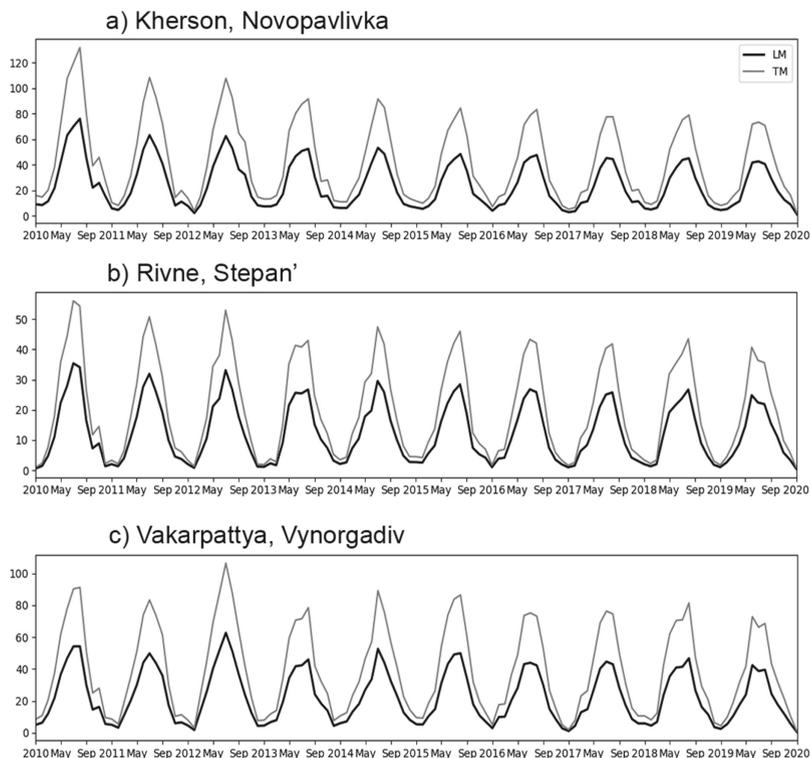


Fig. 5. Simulated amounts of produced CO₂ in kg/ha by sites, summed by months. Black line represents the layered model, gray – the topsoil model results

The plots in Fig. 5 indicate the CO₂ production is significantly lower in the LM simulations. The ratio between the simulated values in LM and TM amounts to 0.562, 0.609 and 0.571 for respective sites over the simulation period. This value is significantly lower than the apparently expected value of 0.8 (equal to the parameter p). The simulated CO₂ production is also lower in the layered model due to reduced carbon stocks in the top soil layer. Thus, a part of SOC that moves down the soil profile is, in a sense, conserved from emitting into the atmosphere.

6 Conclusions

Modeling of soil organic carbon decomposition received much attention from researchers during the last decades. However, the processes behind it are highly complex and depend on multiple environmental factors, and their formal description is far from being exhaustive. Considering multiple soil layers is an important addition, as it introduces processes of downward organic carbon movement, leaching and delayed release of produced carbon dioxide.

To simulate the layered SOC decomposition, we employed RothPC-1 – a layered counterpart of the well-known RothC model. We attempted to overcome a simplification made in the model about the influence of temperature by coupling it with a physically-based soil water and heat flow model. This should be further complemented by choosing and fitting an appropriate temperature effect function.

The presented numerical experiments show the difference between simulated carbon decomposition in the topsoil RothC and layered RothPC-1 models. Even with a high chosen partitioning parameter p , that corresponds to slow downward movement, the differences in simulation results are significant, especially for carbon dioxide emissions.

We also attempted to demonstrate the effect of weather and soil conditions by choosing sites in distinct climatic zones. The results revealed that the same amounts of plant material inputs cause different behavior of SOC on the sites. As expected, the site located in the arid climate corresponded to the highest decomposition rates.

The described model, although a combination of acknowledged good practices, has room for improvement. Further research is required to fit the abiotic functions and validate the model on the measurement data.

References

1. Bashir, O., et al.: Soil organic matter and its impact on soil properties and nutrient status. In: Dar, G.H., Bhat, R.A., Mehmood, M.A., Hakeem, K.R. (eds.) *Microbiota and Biofertilizers*, Vol 2, pp. 129–159. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-61010-4_7
2. Canty, J., Frischling, B., Frischling, D.: Weatherbase (2022). <http://www.weatherbase.com/>
3. Cardinael, R., Guenet, B., Chevallier, T., Dupraz, C., Cozzi, T., Chenu, C.: High organic inputs explain shallow and deep SOC storage in a long-term agroforestry system - combining experimental and modeling approaches. *Biogeosciences* **15**(1), 297–317 (2018). <https://doi.org/10.5194/bg-15-297-2018>
4. Chen, W., Shen, H., Huang, C., Li, X.: Improving soil moisture estimation with a dual ensemble Kalman smoother by jointly assimilating AMSR-E brightness temperature and MODIS LST. *Remote Sens.* **9**(3), 273 (2017). <https://doi.org/10.3390/rs9030273>
5. Cherlinka, V.: Models of soil fertility as means of estimating soil quality. *Geogr. Cassoviensis* **10**, 131–147 (2016)
6. Chui, Y., Moshynskyi, V., Martyniuk, P., Stepanchenko, O.: On conjugation conditions in the filtration problems upon existence of semipermeable inclusions. *JP J. Heat Mass Transf.* **15**(3), 609–619 (2018). <https://doi.org/10.17654/HM015030609>
7. Coleman, K., Jenkinson, D.: RothC-26.3 - a Model for the turnover of carbon in soil, vol. 38, pp. 237–246 (1996). https://doi.org/10.1007/978-3-642-61094-3_17
8. Conant, R., et al.: Temperature and soil organic matter decomposition rates - synthesis of current knowledge and a way forward. *Glob. Change Biol.* **17**(11), 3392–3404 (2011). <https://doi.org/10.1111/j.1365-2486.2011.02496.x>
9. Del Grosso, S., et al.: *Modeling Carbon and Nitrogen Dynamics for Soil Management*, pp. 303–332. CRC Press (2001)

10. Giardina, C., Litton, C., Crow, S., Asner, G.: Warming-related increases in soil CO₂ efflux are explained by increased below-ground carbon flux. *Nat. Clim. Chang.* **4**, 822–827 (2014). <https://doi.org/10.1038/nclimate2322>
11. Hersbach, H., et al.: Operational global reanalysis: progress, future directions and synergies with NWP. ERA Report 27 (2018). <https://doi.org/10.21957/tkic6g3wm>, <https://www.ecmwf.int/node/18765>
12. Hilinski, T.E.: Implementation of exponential depth distribution of organic carbon in the CENTURY model (2001). <https://www2.nrel.colostate.edu/projects/irc/public/Documents/Software/Century5/Reference/html/Century/exp-c-distrib.htm>
13. Ise, T., Moorcroft, P.: The global-scale temperature and moisture dependencies of soil organic carbon decomposition: an analysis using a mechanistic decomposition model. *Biogeochemistry* **80**, 217–231 (2006). <https://doi.org/10.1007/s10533-006-9019-5>
14. Jenkinson, D., Coleman, K.: The turnover of organic carbon in subsoils. part 2. modelling carbon turnover. *Europ. J. Soil Sci.* **59**, 400–413 (2008). <https://doi.org/10.1111/j.1365-2389.2008.01026.x>
15. Kashtan, V., Hnatushenko, V., Zhir, S.: Information technology analysis of satellite data for land irrigation monitoring : Invited paper. In: 2021 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), pp. 1–4 (2021). <https://doi.org/10.1109/UkrMiCo52950.2021.9716592>
16. Kerr, D., Ochsner, T.: Soil organic carbon more strongly related to soil moisture than soil temperature in temperate grasslands. *Soil Sci. Soc. Am. J.* **84**(2), 587–596 (2020). <https://doi.org/10.1002/saj2.20018>
17. Kirschbaum, M.: The temperature dependence of soil organic matter decomposition, and the effect of global warming on soil organic C storage. *Soil Biol. Biochem.* **27**(6), 753–760 (1995). [https://doi.org/10.1016/0038-0717\(94\)00242-S](https://doi.org/10.1016/0038-0717(94)00242-S)
18. Kozhushko, O., Boiko, M., Kovbasa, M., Martyniuk, P., Stepanchenko, O., Uvarov, M.: Evaluating predictions of the soil moisture model with data assimilation by the triple collocation method. *Compu. Sci. Appl. Math.* **2**, 25–35 (2022). <https://doi.org/10.2413-6549-2021-2-03>
19. Kozhushko, O., Boiko, M., Kovbasa, M., Martyniuk, P., Stepanchenko, O., Uvarov, M.: Field scale computer modeling of soil moisture with dynamic nudging assimilation algorithm. *Math. Model. Comput.* **9**(2), 203–216 (2022). <https://doi.org/10.23939/mmc2022.02.203>
20. Krinner, G., et al.: A dynamic global vegetation model for studies of the coupled atmosphere-biosphere system. *Global Biogeochem. Cycles* **19**(1), GB1015 (2005). <https://doi.org/10.1029/2003GB002199>
21. Kroes, J., Roelmsa, J.: ANIMO 3.5; user’s guide for the ANIMO version 3.5 nutrient leaching model. Wageningen, SC-DLO, 1998. Techn. Doc. 46, 98 pp. (1998)
22. Langergraber, G., Rousseau, D., Garcia, J., Mena, J.: CWM1: a general model to describe biokinetic processes in subsurface flow constructed wetlands. *Water Sci. Technol.* **59**(9), 1687–1697 (2009). <https://doi.org/10.2166/wst.2009.131>
23. Lei, J., Guo, X., Zeng, Y., Zhou, J., Gao, Q., Yang, Y.: Temporal changes in global soil respiration since 1987. *Nat. Commun.* **12**, 403 (1987). <https://doi.org/10.1038/s41467-020-20616-z>
24. Li, C., Farahbakhshazad, N., Jaynes, D., Dinnes, D., Salas, W., McLaughlin, D.: Modeling nitrate leaching with a biogeochemical model modified based on observations in a row-crop field in Iowa. *Ecol. Model.* **196**, 116–130 (2006). <https://doi.org/10.1016/j.ecolmodel.2006.02.007>

25. Luo, Z., Luo, Y., Wang, G., Xia, J., Peng, C.: Warming-induced global soil carbon loss attenuated by downward carbon movement. *Glob. Change Biol.* **26**(12), 7242–7254 (2020). <https://doi.org/10.1111/gcb.15370>
26. Parton, W., Ojima, D., Cole, C., Schimel, D.: A general model for soil organic matter dynamics: sensitivity to litter chemistry, texture and management, pp. 147–167. John Wiley & Sons, Ltd. (1994). <https://doi.org/10.2136/sssaspecpub39.c9>
27. Parton, W., Scurlock, J., Ojima, D., Gilmanov, T., Scholes, R., Schimel, D., Kirchner, T., Menaut, J.C., Seastedt, T., Moya, G., Kamnalrut, A., Kinyamario, J.: Observations and modeling of biomass and soil organic matter dynamics for the grassland biome worldwide. *Global Biogeochem. Cycles* **7**(4), 785–809 (1993). <https://doi.org/10.1029/93GB02042>
28. Peralta, G., Di Paolo, L., Omuto, C., Viatkin, K., Luotto, I., Yigini, Y.: Global soil organic carbon sequestration potential map technical manual (2020). <https://fao-gsp.github.io/GSOCseq/index.html>
29. Poggio, L., et. al.: SoilGrids 2.0 : producing soil information for the globe with quantified spatial uncertainty. *Soil* **7**(1), 217–240 (2021). <https://doi.org/10.5194/soil-7-217-2021>
30. USDA NRCS (Natural Resources Conservation Service): Interpreting the soil conditioning index: a tool for measuring soil organic matter trends, no. 16 (2003). https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053273.pdf
31. Shelia, V., Simunek, J., Boote, K., Hoogenboom, G.: Coupling DSSAT and HYDRUS-1D for simulations of soil water dynamics in the soil-plant-atmosphere system. *J. Hydrol. Hydromech.* **66**(2), 232–245 (2018). <https://doi.org/10.1515/johh-2017-0055>
32. Shibu, M., Leffelaar, P., Van Keulen, H., Aggarwal, P.: Quantitative description of soil organic matter dynamics - a review of approaches with reference to rice-based cropping systems. *Geoderma* **137**(1), 1–18 (2006). <https://doi.org/10.1016/j.geoderma.2006.08.008>
33. Shibu, M., Leffelaar, P., van Keulen, H., Aggarwal, P.: LINTUL3, a simulation model for nitrogen-limited situations: application to rice. *Eur. J. Agron.* **32**(4), 255–271 (2010). <https://doi.org/10.1016/j.eja.2010.01.003>
34. Stepanchenko, O., Shostak, L., Kozhushko, O., Moshynskyi, V., Martyniuk, P.: Modelling soil organic carbon turnover with assimilation of satellite soil moisture data. In: *Modeling, Control and Information Technologies: Proceedings of International Scientific and Practical Conference*, pp. 97–99, no. 5 (2021). <https://doi.org/10.31713/MCIT.2021.31>
35. Taghizadeh-Toosi, A., Christensen, B., Hutchings, N., Vejlin, J., Katterer, T., Glen-dining, M., Olesen, J.: C-TOOL: a simple model for simulating whole-profile carbon storage in temperate agricultural soils. *Ecol. Model.* **292**, 11–25 (2014). <https://doi.org/10.1016/j.ecolmodel.2014.08.016>
36. Thea, C.: Lametsy – large meteorological system (2022). <https://lametsy.pp.ua>
37. Wójcik, W., Osypenko, V., Osypenko, V., Lytvynenko, V., Askarova, N., Zhas-sandykyzy, M.: Hydroecological investigations of water objects located on urban areas. *Environmental Engineering V*. In: *Proceedings of the 5th National Congress of Environmental Engineering*, pp. 155–160 (2017)
38. Xie, E., Zhang, X., Lu, F., Peng, Y., Chen, J., Zhao, Y.: Integration of a process-based model into the digital soil mapping improves the space-time soil organic carbon modelling in intensively human-impacted area. *Geoderma* **409**, 115599 (2021). <https://doi.org/10.1016/j.geoderma.2021.115599>

39. Yang, J., Yang, J., Dou, S., Hoogenboom, G.: Simulating the effect of long-term fertilization on maize yield and soil C/N dynamics in northeastern China using DSSAT and CENTURY-based soil model. *Nutr. Cycl. Agroecosyst.* **95**, 287–303 (2013). <https://doi.org/10.1007/s10705-013-9563-z>
40. Yoder, D., et al.: Soil health: meaning, measurement, and value through a critical zone lens. *J. Soil Water Conserv.* **77**(1), 88–99 (2022). <https://doi.org/10.2489/jswc.2022.00042>
41. Zhang, X., Xie, Z., Ma, Z., Barron-Gafford, G., Scott, R., Niu, G.Y.: A microbial-explicit soil organic carbon decomposition model (MESDM): development and testing at a semiarid grassland site. *J. Adv. Model. Earth Syst.* **14**(1), e2021MS002485 (2022). <https://doi.org/10.1029/2021MS002485>
42. Zhou, J., Cheng, G., Wang, G.: Numerical modeling of wheat irrigation using coupled HYDRUS and WOFOST models. *Soil Sci. Soc. Am. J.* **76**, 648–662 (2012). <https://doi.org/10.2136/sssaj2010.0467>
43. Zhou, X., Xu, X., Zhou, G., Luo, Y.: Temperature sensitivity of soil organic carbon decomposition increased with mean carbon residence time: field incubation and data assimilation. *Glob. Change Biol.* **24**(2), 810–822 (2018). <https://doi.org/10.1111/gcb.13994>



Optimization of Coagulant Dosing Process for Water Purification Based on Artificial Neural Networks

Andrii Safonyk[✉] and Myroslav Matviichuk[✉]

National University of Water and Environmental Engineering, Rivne, Ukraine
{a.p.safonyk,m.v.matviichuk}@nuwm.edu.ua

Abstract. The article reveals the essence and features of the neural network model used to regulate water purification processes. The peculiarities, principles and main stages of water purification are determined, which consist of the use of rectification and deposition of separated particles, pre-filtration. It is noted that each process is closely related to the other. The stages of modeling an artificial neural network with descriptions of each step sequentially are revealed. The approach to the use of artificial neural networks during dosing of the mixture for water purification is substantiated. The process of dosing the mixture for water purification and related indicators that are influential for the implementation of the main process are analyzed. A number of factors that directly affect the coagulation process and, as a consequence, the structure of the neural network include turbidity, flow rate and working pumps. However, it is emphasized that there are other parameters, such as pH, conductivity and water temperature, which also have a slow effect on rectification and coagulation, but are not critical. The coagulant regulates turbidity, minimizing the cost of production, which is a very important factor in the economic efficiency of the enterprise. It is emphasized that determining the dose of coagulant is necessary to minimize time, implementation without an intermittent process, stabilize variations in the observations of the operator and improve the quality of the end result. Taking into account the parameters of influence, it is proposed to establish a special regime of coagulant dose control, on the basis of which a control scheme is developed, which is presented schematically with the separation of the main processes occurring during coagulant dosing. Given the complexity of building a neural network model, a neural network model with uncontrolled learning is proposed, which is used to construct multidimensional space in space with a lower dimension to optimize coagulant dosing in water purification.

Keywords: Artificial neural network · Automatic dosing · Coagulant · Water · Purification · Suspended solids · Concentration · Intelligent information system

1 Introduction

An important problem is the rational use of water resources and the minimization of water pollution by industrial complexes. In many production cycles, water can be reused after a purification procedure. In the process of water purification, coagulation is an important element. Thus, in the [2] study, the author investigated the possibility of reusing technical water after undergoing only one stage of purification - coagulation, and concluded that in many cases it is quite an effective solution. Its action is aimed at combining suspended solids present in water into larger particles. Chemical coagulants are added to the water to improve the removal of particles, colloids and solutes. However, an important factor in the quality of water purification is the optimization of coagulation (coagulation) to select the optimal method and dosage of coagulation.

For the fastest and most effective cleansing by coagulation, it is advisable to use coagulant dosing systems based on the neural network. It also avoids over-consumption of reagents or insufficient water purification, as well as quick response to changes in changing water parameters, such as temperature or pH level. Research in this area is quite relevant

The main contribution of this work is the proposed model of neural network with uncontrolled learning, which is used to construct multidimensional space in space with a lower dimension to optimize the dosage of coagulant in the process of water purification.

It is proposed to use an optical sensor to control some water parameters based on the readings of which the neural network adjusts the dosage, which eliminates the need for laboratory testing of water samples. In addition, the process of water purification and coagulant dosing is analyzed, the factors influencing this process are determined.

2 Problem Statement

Today, modern industrial enterprises, whose work is aimed at water purification, in most cases, use the method of determining the dose of coagulant by trial coagulation of water, which aims to determine the level of optimal dosage of coagulant. However, this technique has one significant drawback, it is time consuming, intermittent and subject to variation in the observations of the operator, which is the last link in the end result. Also, frequent performance of this test requires a large number of chemicals for testing, and also requires professional workers to for obtain the most accurate results in determining the required dose of coagulant.

To solve the above problems, it is important to use methods based on artificial neural network (ANN), which are used to control the dosage of coagulant at the water treatment plant. Thanks to which the optimization of coagulants is performed as quickly as possible, efficiently. The dosing rate of the coagulant is nonlinearly correlated with water parameters such as turbidity, conductivity, pH, temperature and the like.

Thus, within the framework of our research, the approach to the use of artificial neural networks during dosing of the mixture for water purification is substantiated. Built a neural network model with uncontrolled learning, which is used to construct multidimensional space in space with a lower dimension to optimize coagulant dosing in water purification.

3 Literature Review

As noted in the Introductory Section, today coagulation is a fairly common and relevant method of water purification. Publications on the use of machine learning to automate the coagulant dosing process in water bodies have been a popular area of research for the past 10 years.

However, in the course of research presented in article [5], it was concluded that with increasing volumes of water in need of purification and decreasing the concentration of impurities in it, this method of purification becomes less efficient and more expensive. Therefore, it is most appropriate to use this method where water is reused, which has already been mentioned in studies [2] above, that also discusses considers the possibility of using only one method of chemical coagulation as a sufficiently effective method of water treatment in industry.

So, in review [1] discusses a wastewater treatment plant and this article discusses a new adaptive control based on an online machine for extreme extreme learning (OS-ELM). The author notes that traditional methods rely heavily on human intervention and cannot adapt to changing environments, whereas neural network systems have advantages in such cases. A similar online neural network learning system with direct communication is also described in the article [4], but took into account more parameters, such as the influence of rainy weather.

Papers [3] present water treatment technology creatively and a predicting model of coagulant dosing rate was established. Based on the test results of the developed model, the author of the article concluded that the dosage of coagulant using neural networks are effective. In articles [9] the modeling of artificial neural networks for water purification with low level of pollution is performed. It is concluded that the use of such models of neural networks is also effective for predicting the level of substances in water, such as nitrogen concentration in effluents

So, after analyzing these articles, we can conclude that such dosing systems are quite effective solutions, and the use of neural networks, not only to determine the dose of coagulant, but to predict water parameters, such as the method that demonstrated in works [6–8] for the construction of an intelligent system for determining the concentration of iron in the coagulant by its color on the basis of a neural network is considered, can significantly optimize this process.

However, given the described scientific findings on the topic, the issue of disclosing the principles of neural network control for coagulant dosing in the process of water purification remains open and requires detailed study. In the reviewed literature, the authors did not consider the variant of the neural network model proposed in this paper.

4 Materials and Methods

The main parameters of quality purified raw water are turbidity, electrical conductivity, pH, temperature and others. Given the high probability of uncertainty and variability of the initial input data, an important requirement for the water treatment system is its stability. Based on the fact that water treatment involves biological, physical and chemical processes, the application of the principles of automated coagulation is an undeniable advantage.

4.1 Water Purification Process

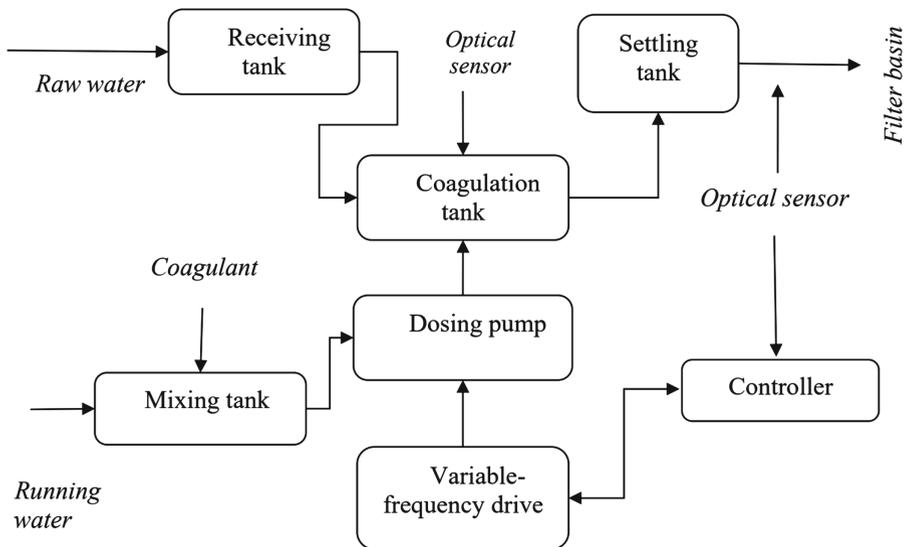


Fig. 1. Schematic diagram of coagulation

The general process of water purification takes into account the processes of disinfection, coagulation, filtration, clarification and pH regulation. The coagulation process is shown in Fig. 1.

When raw water passes through a water pump, its turbidity and velocity are measured in front of the coagulation tank. In addition, after the coagulant and running water are mixed in the appropriate proportions, the mixture is pumped into the coagulation tank and carries out the coagulation reaction with raw water. Coagulation is performed using aluminum sulphate coagulant $Al_2(SO_4)_3$, which results in the process of volumetric sludge formation, which electrochemically binds colloidal particles and solids present in raw water. Dosage of coagulant will take effect in about 30-60 min, which indicates a fairly long coagulation process. The precipitate formed, deposited on the bottom of the

tank, is removed, and the water is tested at the outlet of the tank, measuring the turbidity and assessing whether its value reach the set values. The key point is that the required data is selected by the equipment, calculated by the controller, and the optimal coagulant for the operation of the dosing pump. The pump regulates speed and opening to optimize coagulant dosing online. It is known that the process of coagulant dosing is complex, diverse, nonlinear, with a large delay. It cannot find a direct link between input and output information. Therefore, establishing the correct mathematical model is a difficult task. The next step is the filtration process, which is responsible for removing the contaminant particles left after coagulation precipitation, then ozonation to separate micro-contaminants and chlorination to disinfect and adjust the pH.

The coagulation process is responsible for removing most unwanted substances from raw water, so the main condition is monitoring and control of the process.

4.2 Determination of Coagulant Dose

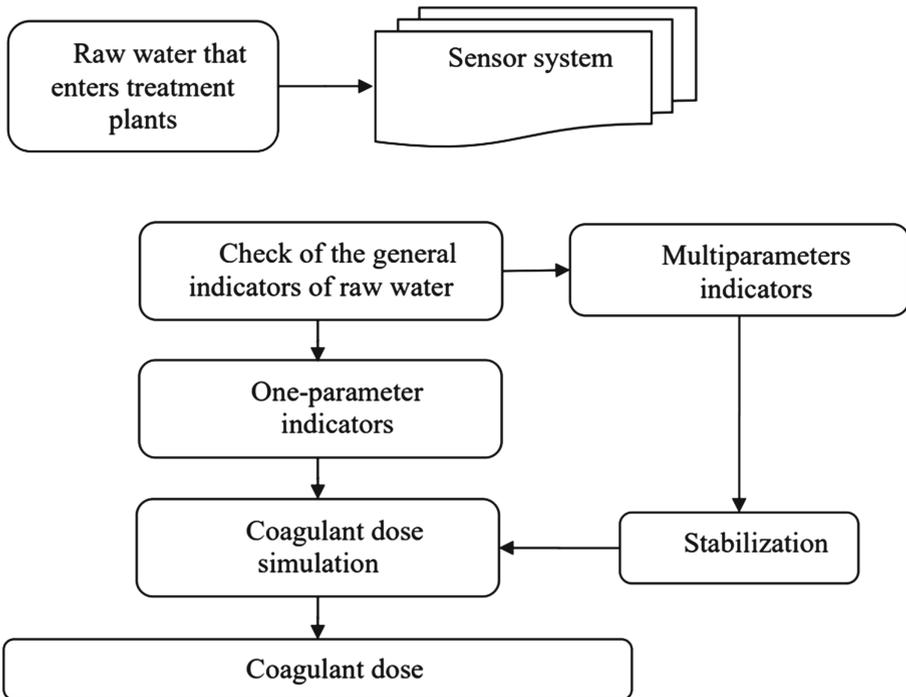


Fig. 2. Stages of determining the dose of coagulant in the coagulation process

Determining the dose of coagulant is necessary to minimize time, implementation without intermittent process, stabilize variations in the observations of the operator and improve the quality of the final result.

The general system for determining the dose of coagulant in the coagulation process consists of three main successive stages:

1. verification of one-parameter data;
2. stabilization of many parametric data;
3. determination of coagulant dose.

4.3 Modeling of Artificial Neural Network

Currently, there are three common modeling methods, namely, the mechanism model, the knowledge-based model and the data-driven model. First, the mechanism model depends very much on the internal mechanism of the system. However, the system becomes more nonlinear and strong in connection, earlier; in coagulation, the internal mechanism is complex and cannot be shown clearly and clearly mathematically justified. Second, the knowledge-based model increases on the limited knowledge of experts with poor adaptability. Finally, the data-driven model does not have data on the internal state of the system; however, such a model can predict and make an informed decision with a large number of data analysis results and reasonable conclusions.

The considered neural network is aimed at performing tasks in conditions of uncertainty of parameters, this type of task is the process of dosing coagulant without analyzing its mechanism, collected insufficient and inaccurate expertise, variability of input parameters and indicators.

Simulation of an artificial neural network occurs in three stages:

The first stage is the stage of model construction. This stage is the main in the formation of the structure, it determines the number of nodes, layers and establishes connections between network nodes.

The second stage is the network learning phase. Neural network training is carried out by changing the weights of the nodes in accordance with the chosen learning algorithm (learning with a teacher, without a teacher and blended learning). The learning process is divided into two stages: training and testing. Implementing network learning involves minimizing the value of the error function by changing the values of the weights between neurons and layers.

The third stage is the network testing stage. Neural network testing is aimed at identifying the level of network quality and its ability to generalize the end result. The effective operation of the neural network is based on minimizing the difference between the predicted and actual values of the original data. The basis of factors that directly affect the process of coagulation of raw water and, as a consequence, the structure of the artificial neural network include:

- turbidity
- flow rate,

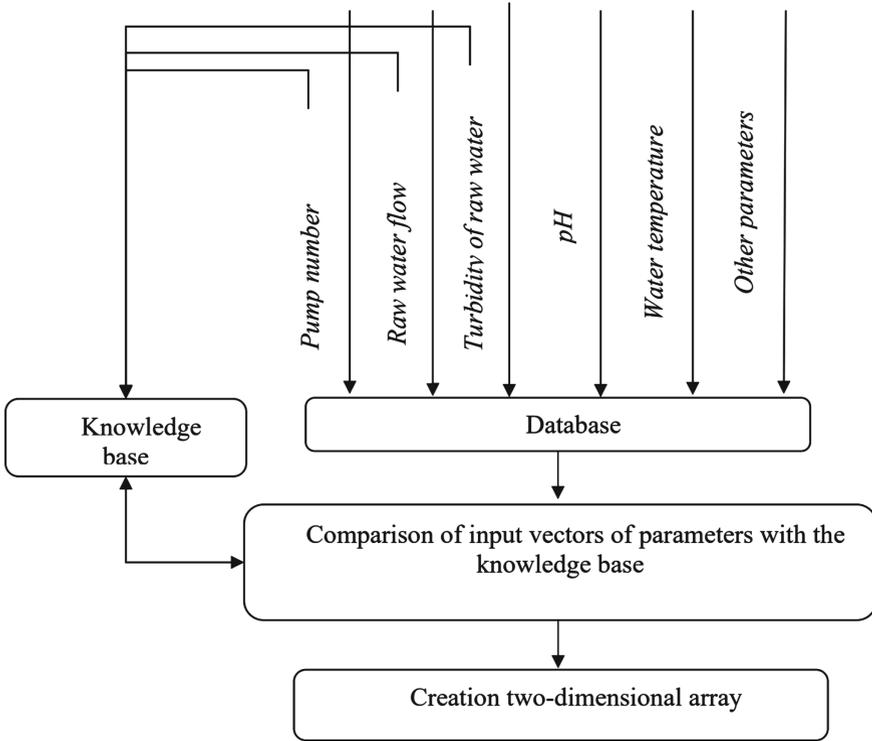


Fig. 3. Scheme of formation of a two-dimensional array of neural network nodes

Other parameters, such as *pH*, conductivity and water temperature, also have an effect on coagulation, but are not critical.

The process of the system is implemented online when the untreated water entering the treatment plant enters the sensor base, where there is work with various parameters of input data, their verification. In the case of one-parameter data, the network assigns a reliable value of one, otherwise zero. Multiparameter data undergoes the process of stabilization and sequential assignment of values. Given the uncertainty and instability of the incoming data, it is proposed to use a neural network with uncontrolled learning, which is used to construct multi-dimensional space in space with a lower dimension (usually two-dimensional).

This approach will allow the parameters of raw water quality, which act as the entrance space to be presented in the form of a two-dimensional array of network nodes, shown in Fig. 3 .

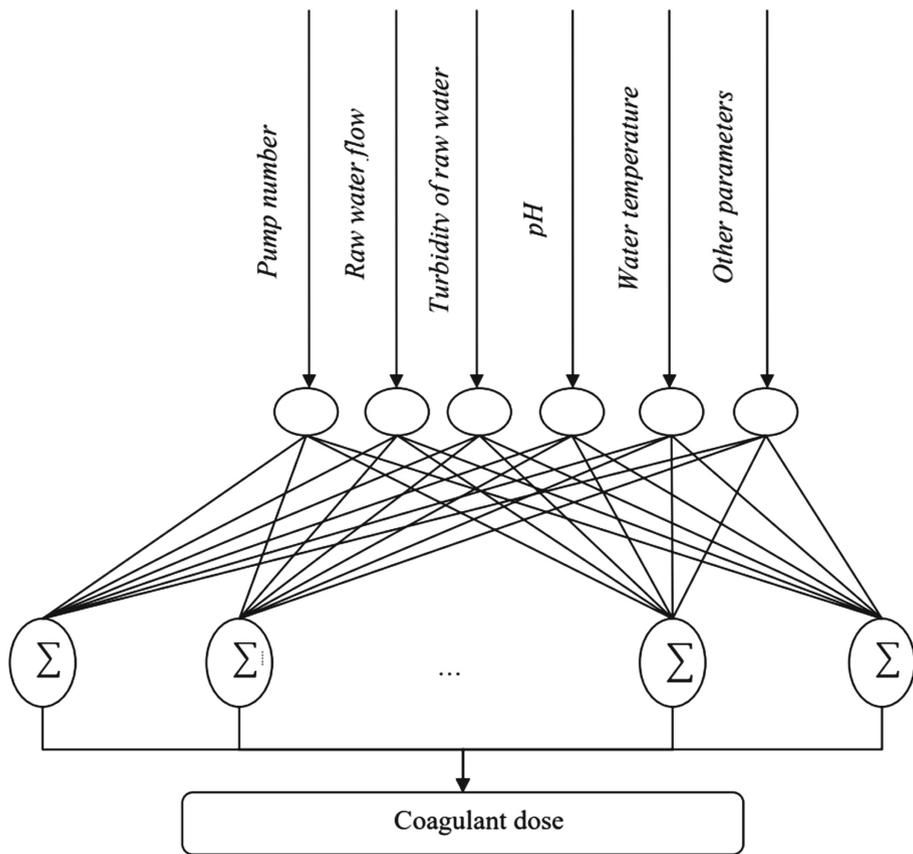


Fig. 4. Neural network model for coagulant dosing in water purification process

In order to maximize the time constant, we apply the Gaussian Filter:

$$K(i) = \exp\left(-\frac{1}{2\sigma_i^2} \|x - m_i\|^2\right) \tag{1}$$

In these expressions σ_i^2 are the value indicating the degree of influence of the i -th parameter; x is the vector; m_i is the network node; $\|x - m_i\|$ is the vector distance between network nodes.

The method of k -spanning neighbors is used then increase accuracy of stabilization of parameters:

$$x(j) = \frac{\sum_{i=1}^k K(m_i)(j)}{\sum_{i=1}^k K(i)} \tag{2}$$

In these expressions $m_i(j)$ is the value of the network node j according to the prototype i .

Thus, the proposed approach will allow to present the parametric components of water flow in two dimensions, atypical measurements such as pollution, non-compliance with regulations, etc., can be detected by measuring the distance between each input vector and its nearest reference vector. Simulation of the neural network for coagulant dosing in the process of water purification is based on the generated two-dimensional data set using a multilayer perceptron with sigmoid activation functions, the scheme of implementation of this process is shown in Fig. 4

In the process of forming the principle of coagulant dosing for water purification, the obtained neural network model is learning using the inverse error propagation algorithm, the next step is to assess the relative importance of weights in the process minor parameters.

5 Experiment, Results and Discussion

Experimentally, with the help of an optical sensor, we will check the turbidity of water at different concentration of suspended solids for 60 min. The optical sensor of control of turbidity of water is based on the principle of measurement of a light stream through solution (water). Turbidity of water is due to the presence of various minerals, fine impurities, insoluble inorganic compounds. When studying the level of turbidity of water, the optical sensor detects the intensity of transmitted light and compares it with the reference measurements of pure water. During the study, several samples were taken and the relationship between the color of the solution was constructed. This dependence is shown in Fig. 6.

Analyzing these dependencies of color components and concentration, it can be concluded that these parameters show stable correlations. Execution of the optical sensor for determination of concentration allows to get rid of a stage of laboratory analysis of water and to accelerate reaction of system to change of parameters of water. However, the use of this method may be inaccurate in the presence of some individual pigments in the water, so it requires further research.

The obtained data are used to evaluate the properties of water, and transmitted to the neural network to predict the concentration of suspended solids concentration in water and subsequently to determine the dose of coagulant. With each new measurement, the accuracy of neural network prediction, and hence the accuracy of coagulant dosing, increases, as shown in Fig. 7.

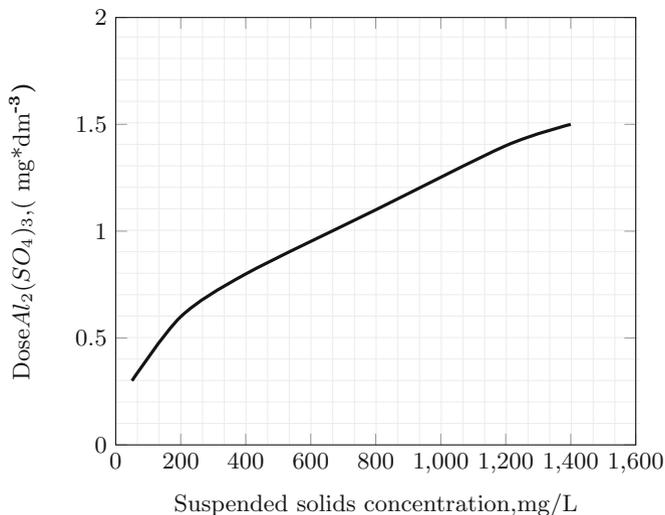


Fig. 5. Regulatory dosage of coagulant from concentration of suspended solids in water.

From Fig. 7 it can be concluded that the disadvantage of the system is some initial period of time during which the system does not always respond accurately to changes in water parameters, and therefore the determined dose of coagulant is not optimal. Over time, this discrepancy decreases in the optimal dosage is achieved.

From the experiment performed during the operation of the proposed neural system, the determination of the coagulant dose within the coagulation process of determining the main parameter for determining the coagulant dose was much faster. This is shown in Fig. 8, which shows the residue of suspended particles in water after the purification step for the same period of time for samples with different initial impurity content at manual dose selection compared to neural network dosing. Reducing time helps increase productivity. There is also a clear predominance of the neural system to determine the concentration of suspended particles in the water subjected to the purification process. The latter fact improves efficiency in the accuracy of determining the level of coagulant in the coagulation process, which positively contributes to the quality of the final product.

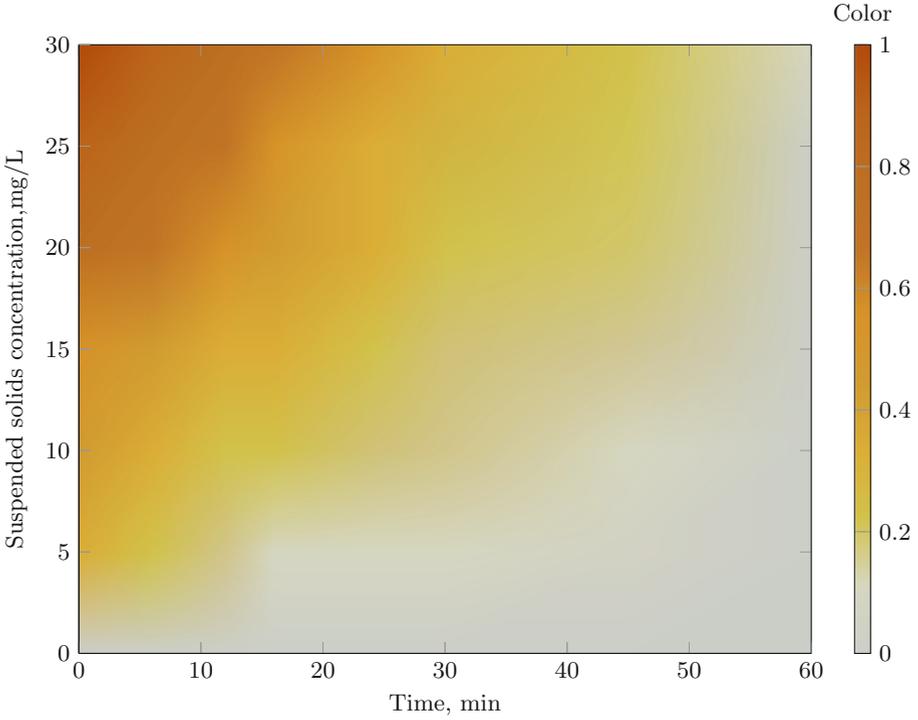


Fig. 6. Dependence color of water on time and of suspended solids concentration

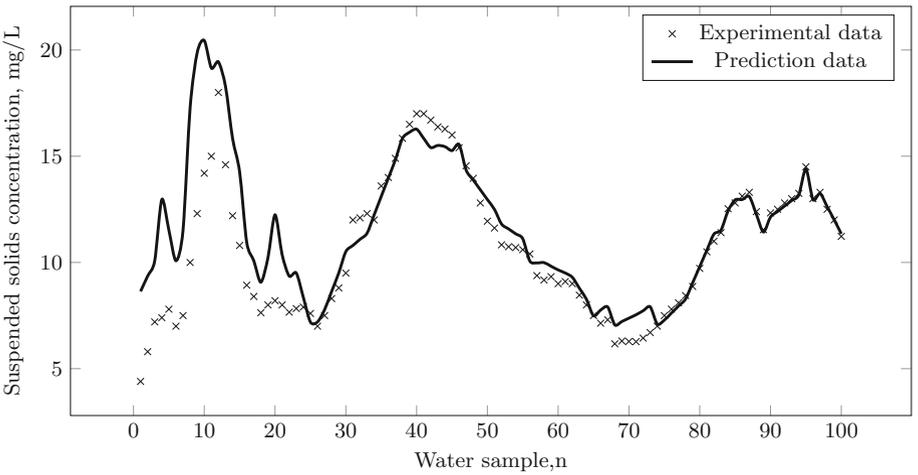


Fig. 7. Comparison of experimental data and data obtained from the model using the neural network

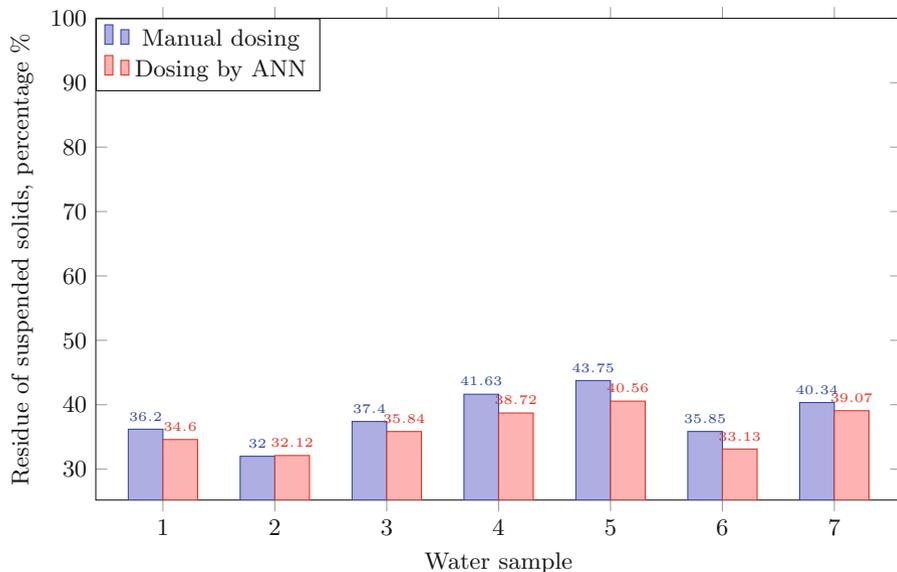


Fig. 8. Comparison of water purification efficiency

Execution of the optical sensor for determination of concentration allows to get rid of a stage of laboratory analysis of water, and to accelerate reaction of system to change of parameters of water.

Joint use with the neural network for predictive determination of other water quality parameters is promising. This will reduce the reaction time of the system and improve the accuracy of coagulant dosing.

6 Conclusions

The paper reveals the principles of neural network control for coagulant dosing in the process of water purification. Features, principles and main stages of water purification are determined. The process of dosing the mixture (coagulant) for water purification and related indicators that are influential for the implementation of the main process are analyzed and related factors. Given the complexity of building a neural network model, the recommended neural network model with uncontrolled learning is used to construct multidimensional space in a lower dimension space to optimize coagulant dosing during water treatment. The proposed neural network model has a variable structure. Since the number of hidden level nodes is always a key point for modeling, the neural network model is not fixed here when the analysis of the main components for latent level universality is implemented. The proposed method can effectively solve the problem of re-equipment and reliability of neural networks and increase the speed of calculation and generalization. Experimental data collection was performed and compared with the data obtained using the neural network.

References

1. Cao, W., Yang, Q.: Online sequential extreme learning machine based adaptive control for wastewater treatment plant. *Neurocomputing* **408**, 169–175 (2020). <https://doi.org/10.1016/j.neucom.2019.05.109>
2. Da Silva, L.F., Barbosa, A.D., de Paula, H.M., Romualdo, L.L., Andrade, L.S.: Treatment of paint manufacturing wastewater by coagulation/electrochemical methods: proposals for disposal and/or reuse of treated water. *Water Res.* **408**, 467–475 (2016). <https://doi.org/10.1016/j.watres.2016.05.006>
3. Hua, B., Gao, Lixin Li, G.: Neural networks based optimum coagulation dosing rate control applied to water purification system. *Eur. J. Cardiovasc. Nur.* **2**, 1432–1435 (2002). <https://doi.org/10.1109/WCICA.2002.1020819>
4. Qiao, J., Han, H., Han, H.: Neural network on-line modeling and controlling method for multi-variable control of wastewater treatment processes. *Mater. Today Proc.* **16**, 1213–1223 (2013). <https://doi.org/10.1002/asjc.758>
5. Rahman, M.S., Islam, M.R.: Sustainability of current water purification technologies. *Sustainable Water Purification*, pp. 59–122 (2020). <https://doi.org/10.1002/9781119651123>
6. Safonyk, A., Mishchanchuk, M., Hrytsiuk, I.: Spectrophotometric method for coagulant determining in a stream based on an artificial neural network. In: Babichev, S., Lytvynenko, V. (eds.) *ISDMCI 2021. LNDECT*, vol. 77, pp. 589–601. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-82014-5_40
7. Safonyk, A., Mishchanchuk, M., Lytvynenko, V.: Intelligent information system for the determination of iron in coagulants based on a neural network (2021). <http://ceur-ws.org/Vol-2853/short13.pdf>
8. Safonyk, A., Tarhonii, I., Hrytsiuk, I., Rudyk, A., Ilkiv, I.: Dynamic optimization of the technological water treatment process automatic control system. In: *Conference Paper 11th International Conference on Advanced Computer Information Technologies (ACIT2021)*, Deggendorf, Germany, pp. 13–18 (2021). <https://doi.org/10.1088/10.1109/ACIT49673.2020.9208814>
9. Wang, S., Li, Q., Zhao, C., Zhu, X., Yuan, H., Dai, T.: Denitrification mechanism and artificial neural networks modeling for low-pollution water purification using a denitrification biological filter process. *Sep. Purif. Technol.* **257**, 117918 (2021). <https://doi.org/10.1016/j.seppur.2020.117918>



Methodology for Solving Forecasting Problems Based on Machine Learning Methods

Irina Kalinina  and Aleksandr Gozhyj 

Petro Mohyla Black Sea National University, Nikolaev, Ukraine
alex.gozhyj@gmail.com

Abstract. The article presents a methodology for solving forecasting problems based on machine learning methods. The technique consists of the following stages - data collection stage; research stage and data preparation; model training stage; the stage of determining the effectiveness of the model; stage of improving the efficiency of the model; visualization. The methodology is designed to systematically solve forecasting problems using various machine learning methods. Using the methodology as an example, the solution of the problem of forecasting the production of electricity for combined power plants using regression, tree-based models, boosting, bagging and neural networks is considered. A feature of the technique is the definition and study of the effectiveness of machine learning models and their improvement. The efficiency of predictive models of various types is analyzed. Various quality indicators were used to assess the quality of forecasts. Boosting is the best predictive model. The results obtained indicate the good quality of the predicted values using the proposed methodology.

Keywords: Forecasting methodology · Machine learning · Regression · Decision trees · Bagging · Boosting · Neural networks · Model quality criteria

1 Introduction

Machine learning is an effective system tool for building and developing predictive models based on experimental data, as well as for deep analysis and efficient solution of large-scale projects based on various data processing technologies [4, 6, 11, 13, 14, 17–19, 25]. Machine learning is based on methods, procedures and algorithms for transforming various data sets into reasonable actions. A significant increase in data volumes required additional computing power, which in turn stimulated the development of statistical methods for analyzing large data sets.

One of the main tasks of machine learning is prediction. Forecasting is implemented on the basis of models built and trained on experimental (statistical) data using certain procedures. Depending on the type of training (with a teacher,

without a teacher, with reinforcement), the composition of these procedures is different. The purpose of forecasting is to improve the decision-making process under conditions of uncertainty. To achieve this, forecasts must give an unbiased guess about what is likely to happen (point forecast) and also define a measure of uncertainty, such as the forecast interval.

To date, the specialized literature describes a large number of examples of applications of forecasting methods implemented on structured and unstructured data sets [4, 10, 16]. The most common among them are autoregression (AR), ARMA, integrated moving average autoregression (ARIMA), linear and non-linear multiple regression, quantile regression, regression trees, neural networks, Bayesian neural network methods that are applied and give good results [11, 13, 14, 18, 19]. The main problems of machine learning procedures - overfitting and underfitting, are associated with the presence of noise/outliers, gaps, and the presence of non-linearity in real data. In addition, in the tasks of processing time series as initial data, the following problems arise: lack of response to a change in trend, non-linearity, non-stationarity of processes described by time series.

When solving complex machine learning problems, a multi-stage prediction process is applied [5, 14, 19, 22, 25]. Such a systemic multi-stage process makes it possible to improve the forecast by applying new approaches to the aggregation of information and methods, as well as by assessing the quality at each stage of forecasting.

When solving the problem of forecasting, it is necessary to systematically apply methods of research, data preparation and training, as well as procedures for evaluating the effectiveness of models and procedures for improving the efficiency of models by evaluating models and choosing the most effective ones.

Recently, a significant number of works have appeared on the application to solve forecasting problems using neural networks and various statistical methods, such as Bayesian forecasting and complex regression models. However, the prediction accuracy is not always satisfactory [4, 7, 9, 10]. In this case, combined forecasts are used to improve the forecasting accuracy.

Combining or aggregating forecasts is not a new idea, but it has recently received increased attention due to the fact that such an approach gives an increase in the quality of forecast values. Its essence lies in the combination of various predictive methods, since it is difficult at first to determine which of them will be the most effective, and the choice of the most accurate one. This is consistent with the Bayesian approach, as it can be thought of as an update, with each individual prediction added to a combined prediction (also called an ensemble), which improves the performance of the overall prediction.

Combined forecasts are more effective when the forecasts are based on different methods in order to improve the overall forecast performance. Among the most promising, which are suitable for use in combined forecasts, we should single out such methods as bagging and boosting. These ensemble methods are of particular interest if they are based on decision trees. A decision tree is a kind of tree structure used in regression and classification, also called a regression tree

used for prediction. This allows you to extract the rules and refine the non-linear relationship between inputs and outputs easily. The most important advantage of regression trees is the ability to be easily developed and interpreted due to their non-parametric design.

Bootstrap aggregated (bagging) is an ensemble method that can solve the data glut problem for a regression tree. It builds multiple trees iteratively based on resampling with replacement and then integrating them into a voting decision tree to get better accuracy. Despite the presence of new approaches in the field of forecasting, traditional methods of regression analysis, methods based on decision trees, the neural network approach are quite reliable and are not so prone to complicating predictive models. They are applicable in ensemble methods, but only in combination with more sophisticated methods.

Thus, it is of interest to create a systematic approach to solving predictive problems based on machine learning methods, which uses procedures for checking the effectiveness and improving the quality of predictive solutions [14, 20].

Statement of the Research Problem. Develop and study a forecasting methodology based on methods and models of machine learning. Explore various indicators of the quality of forecasting models. Conduct a comparative analysis of the effectiveness of forecasting methods.

2 Forecasting Methodology

Development of forecasting methodology based on machine learning methods allows to solve forecasting problems through the systematic use of various methods and technologies [3–5, 7, 14, 19]. The methodology is based on the principles of a systems approach and consists of the following stages: data collection stage; stage of research and data preparation; stage of model learning; stage of determining the effectiveness of the model; stage of increasing the efficiency of the model; additional visualization procedure that accompanies each of the stages (Fig. 1). The main feature of the methodology is the availability of procedures for evaluating the effectiveness of the model and procedures for improving the effectiveness of the model, which is further evaluated and selected the most effective of the alternatives.

The Data Collection Stage is the stage at which data is collected and downloaded. Procedures for analyzing the structure of the data set are used and the types of features in the set under study are analyzed. If necessary, recoding of signs is carried out. The result of this step is the analyzed data set.

Stage of Research and Data Preparation. At this stage, descriptive statistics of signs are determined. Non-numeric and missing values in the data set are detected. The necessary signs are chosen. Next, we study the relationships between variables. The final procedure of the data preparation stage is the procedure of normalization or standardization of data. The result of the stage is a set of data prepared for modeling.

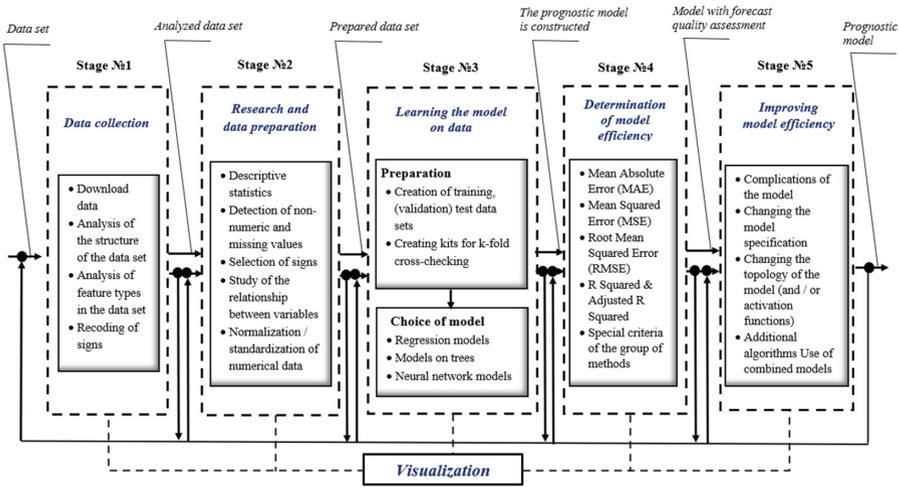


Fig. 1. Forecasting methodology

The Stage of Learning the Model on the Data. The stage consists of two parts: preparation of data samples and selection and training of the model. Training and test data sets are determined during preparation. And also sets for cross-checking are created. The second part compares and selects the most effective prognostic models (regression, tree models or neural networks). The result of the stage is a predictive model.

The Stage of Determining the Effectiveness of the forecast model allows to determine and evaluate using various quality indicators (mean absolute error, root mean square error, correlation, etc.).

At *The Stage of Improving the Efficiency* of the model there are procedures to increase the efficiency of the model (complications, changing the specification and topology of the model, the use of combined models). This is the final stage of the methodology, the result of which is the prognostic model.

The *R* programming language and the *RStudio* development environment [14, 15, 23, 25] are used to implement the software modules.

3 Implementation of the Methodology

3.1 Implementation of Regression Models

Stage 1. Data Collection. The Combined Cycle Power Plant dataset [1] from the UCI machine learning repository is used for forecasting. This data set is used to analyze and forecast the combined cycle power plant’s output based on data collected during 2006–2011, taking into account environmental factors.

This set consists of 9568 observations, each of which has 4 attributes and a label. Observations collected from combined cycle power plants for 6 years, when the power plants were set to work at full load.

The list of all attributes is given in the order in which they are placed in the data set with the original name of each attribute.

1. Temperature (*AT*) in the range of 1.81° C and 37.11° C.
2. Ambient pressure (*AP*) in the range of 992.89 – 1033.30 millibars
3. Relative humidity (*RH*) in the range from 25.56% to 100.16%.
4. Exhaust vacuum (*V*) in the range of 250.36 – 810.56 mm Hg. Art.
5. Net hourly output of electricity (*PE*) 420.26 – 495.76 MW.

Compared to baseline surveys and the ability to perform statistical tests 5×2 times, the data are mixed five times. For each mixing, a 2-fold CV (*cross-validation*) procedure is performed and the obtained 10 measurements are used for statistical testing.

Stage 2. Research and Data Preparation. The data in the set must be in the form that the algorithm can use to build the model. The regression model requires that all features be numeric. To do this, the data were analyzed and their structure determined.

Dependent variable in the model – *PE*, net hourly output of electricity. Before building a regression model, it is necessary to check the normality of the data. For linear regression, the dependent variable does not necessarily have to have a normal distribution, but often the model works better when this condition is met.

Descriptive statistics for the variable *PE* show that the mean is greater than the median. This means that the distribution of net hourly energy output has a shift to the right. Most values are in the range of 430 to 450 MW, but the tail of the distribution extends far beyond these peak values. Such a distribution is not ideal for linear regression, but the fact that this fact is known in advance may help to develop a more appropriate model later.

Study of relationships between traits: correlation matrix. Before adjusting the regression model according to the data, it is determined how the independent variables are related to the dependent variable and to each other. This is determined using the correlation matrix (Fig. 2).

	AT	V	AP	RH	PE
AT	1.0000000	0.8441067	-0.50754934	-0.54253465	-0.9481285
V	0.8441067	1.0000000	-0.41350216	-0.31218728	-0.8697803
AP	-0.5075493	-0.4135022	1.00000000	0.09957432	0.5184290
RH	-0.5425347	-0.3121873	0.09957432	1.00000000	0.3897941
PE	-0.9481285	-0.8697803	0.51842903	0.38979410	1.0000000

Fig. 2. Correlation matrix

In the dataset analyzed, the correlations between *V* and *AP*, *V* and *RH* indicate a rather strong negative linear relationship. Other values of correlations in the matrix indicate a strong intensity of interaction between pairs of corresponding features.

Visualization of Relationships Between Features is carried out using scatter plots. The extended matrix of scattering diagrams for the data set features is shown in Fig. 3. The results of the analysis of scattering diagrams confirm the conclusions obtained after the analysis of the correlation matrix. The histograms, which are located on the diagonal, show the distribution of values of each feature. *LOESS*-curves are additionally added below the diagonal. They indicate a general relationship between the symptoms. Part of the curves resembles the letter “U”, and part “S”, indicating a nonlinear relationship between the features. This conclusion could not be made on the basis of correlations alone. And the *LOESS*-curves for *AT* and *AP*, *AT* and *PE*, *V* and *AP*, *V* and *RH*, *RH* and *PE* are straight lines with different angles.

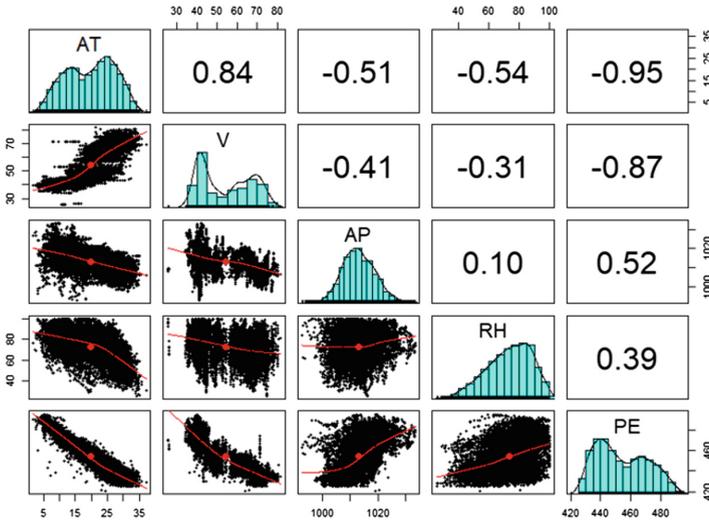


Fig. 3. Extended scattering matrix for data set features

Stage 3. Learning the Model on the Data. The initial phase of learning the model on the data is the procedure of dividing the set into two parts, for training and testing. The data set was divided so that the training data was 75% and the test data was 25% of the original set. Test data are used to evaluate the effectiveness of alternative models.

The first of the alternative regression models is the simple multiple regression model (conventionally denoted by M_R1). The multiple regression model looks like this:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_ix_i + \varepsilon,$$

where ε is the error value added to mention that no prediction is perfect.

Taking into account the notations that are accepted in the data set, we have a model of the form:

$$PE = \beta_0 + \beta_1 \times AT + \beta_2 \times AP + \beta_3 \times V + \beta_4 \times RH.$$

The results obtained using a linear regression model show a logical chain: an increase in temperature (T), relative humidity (RH) and exhaust vacuum (V) leads to a decrease in net time output (PE) and an increase in atmospheric pressure (AP). Leads to an increase in the projected net hourly output of electricity (PE). The result of the stage is the estimation of the parameters of the regression model.

Stage 4. Determining the Effectiveness of the Model. Estimates of the parameters obtained in the previous step show how the independent variables are related to the dependent variable, but do not show how well the model fits the data [2,3,14,19].

From the analysis of the simulation results it is obvious that there are the following ways to evaluate the effectiveness of the model.

1. Summary statistics of forecast errors. Since the residual value is equal to the difference between the true and predicted values, the maximum error is 17.868 and the minimum error is (-43.530). This suggests that at least one observation model overestimated the projected value of net hourly power output (PE) by 43.5 MW. On the other hand, 50% of the errors relate to the values of the 1st and 3rd quarters (first and third quarters), so most forecasts are in the range of 3.15 MW more than the true value and 3.17 MW less than the true value.
2. For each calculated regression coefficient, a p -value is determined, which is an estimate of the probability that the true coefficient for this estimate is zero. Small p -values suggest that the true coefficient is unlikely to be zero, so it is unlikely that this feature is not related to the dependent variable. In the model under consideration, all variables are significant and result-related.
3. The coefficient of determination assesses how well the model generally explains the value of the dependent variable. Since the coefficient of determination in the model is 0.9303, this means that the model explains 93% of the changes in the dependent variable.

The three performance indicators in Fig. 4 show that the presented model works well. However, you can increase the efficiency of the model, described the model in another way.

Stage 5. Improving the Efficiency of the Model. The main difference between the regression model and other machine learning methods is that regression usually leaves the user with the choice of features and specification of the model. Therefore, knowing how the feature is related to the result, you can use this information in the model specification and possibly increase its efficiency.

In the simulation process, the following improvements were made to a simple multiple regression model:

- added a nonlinear variable value of the vacuum in the exhaust;
- added a nonlinear variable for atmospheric pressure;
- the dependence between temperature and atmospheric pressure is introduced.

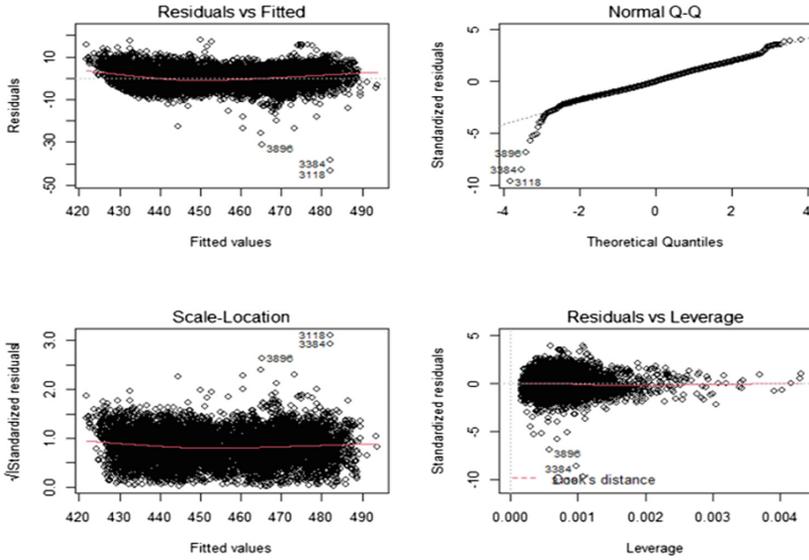


Fig. 4. Diagnostic diagrams for regression of hourly power output

After adding additional variables, the improved M_{R2} model looks like this:

$$PE = \beta_0 + \beta_1 \times AT + \beta_2 \times V + \beta_3 \times V^2 + \beta_4 \times AP + \beta_5 \times AP^3 + \beta_6 \times RH + \beta_7 \times AT \times AP.$$

To select the best of the regression models for each of them, the quality criteria are summarized in Table 1. The models were evaluated by comparing the adjusted coefficients of determination ($Adj.R^2$), residual standard error (SEE), F -statistics and p -values.

Table 1. Comparative table of quality criteria of alternative regression models

Type of regression model	Adjusted coefficients of determination, $Adj.R^2$	Sum of squared error, SSE	F -statistics	p -value
M_{R1}	0,930	4,507	$2,395 \cdot 10^{-4}$	$2,2 \cdot 10^{-16}$
M_{R2}	0,932	4,450	$1,407 \cdot 10^{-4}$	$2,2 \cdot 10^{-16}$

For further forecasting, the M_{R2} model was chosen, which has the best quality indicators in total.

3.2 Implementation of Models Based on Trees

Decision Trees. Stage 1. Data Collection. A previous *Combined Cycle Power Plant* [1] dataset is used to create a predictive model using tree-based models.

Stage 2. Research and Data Preparation. Compared to other types of machine learning models, one of the advantages of trees is that they are able to process different types of data without pre-processing. This means that you do not need to normalize or standardize the symptoms. Although trees are very resistant to unreliable data, it is always advisable to check the data for serious problems. Based on a preliminary analysis of the data set, it is assumed that the data is reliable.

Stage 3. Learning the Model on the Data. The first step to the stage of learning a model on data is to prepare data for the creation of models by dividing the set into training and test. This is done in the proportion of 75% training, 25% test.

For each node in the tree, the number of examples that reach the decision point is specified. A more detailed description of the compliance of the tree, including the root mean square error for each of the nodes and the overall degree of importance of the features, is obtained from the full report on the model. All predictors used in the model, according to the report, are ranked in order of importance.

It's easier to use visualization to understand how a tree works. For the constructed regression tree model, the tree diagram is presented in Fig. 5. Such visualizations help to disseminate the results of the construction of regression trees, because they are easy to understand. The numbers shown in the end nodes are predicted values for the examples reaching the node. Thus, the chart can identify key factors that allow you to get high values of net hourly power output (*PE*), for the [1] dataset.

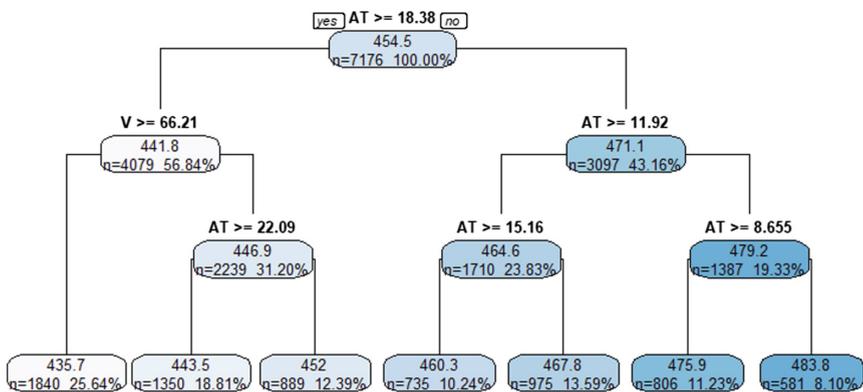


Fig. 5. Visualization of the regression tree model

Stage 4. Determining the Effectiveness of the Model. The constructed regression tree model is used for forecasting based on test data. If we examine the consolidated statistics of the obtained forecasts, the potential problem is determined: the forecasts have a much narrower range than the real values. This result indicates that the model incorrectly identifies extreme cases, in particular, the minimum and maximum values of net hourly power output. However, between the first and third quartiles, it works almost flawlessly.

To check the quality of the forecasts for the regression tree model, the values of the quality metrics were calculated. The correlation between projected and actual power output values provides an easy way to measure model efficiency. The correlation value is 0.95, but it only shows how strongly the predictions are related to real values, but not how far the predictions of real values are.

Another way to assess the effectiveness of the model is to determine how far the average forecast is from the true value, is the calculation of the mean absolute error (*mean absolute error*, MAE). Calculations show that, on average, the difference between the regression tree model predictions and the actual net hourly outputs is approximately 4.2 MW. With a difference between the maximum and minimum values in the test data set of 75 MW, this suggests that the model works quite well. The average quality score for training data is 454.49. If we predicted 454.49 for all samples of test data, we would get an average absolute error of about 14.83. The constructed regression tree (MAE = 4.2) is on average closer to the true value than the conditional average (MAE = 14.83), and much more.

Stage 5. Improving the Efficiency of the Model. To improve the quality of forecasts, a regression tree pruning procedure was performed. Figure 6 presents a graph of the change in the total forecast error from the number of regression tree nodes.

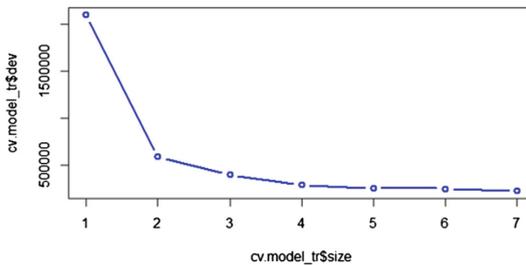


Fig. 6. Graph of change of total forecast error from the number of nodes of the regression tree

In this case, cross-checking selects the most complex tree. However, to create an alternative model, the tree is reduced to 5 end nodes. Visualization of the truncated regression tree model is presented in Fig. 7.

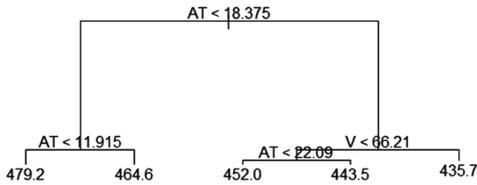


Fig. 7. Visualization of the abbreviated regression tree model

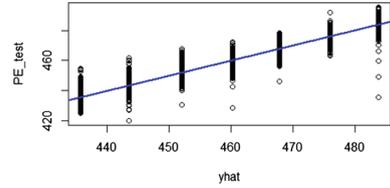


Fig. 8. Estimated values of net hourly power output on the test data set using the regression tree model

According to the results of cross-checking, an uncut tree was used to obtain predictions. The forecast is performed on a test sample of data (Fig. 8).

Thus, the MSE in the control sample of this regression tree is 29.3. The square root of this MSE is approximately 5.41. This means that the predictions of this model deviate from the true value of net hourly electricity output by an average of 5.41 MW.

Bagging and the Random Forest. *Bagging*, or *Bootstrap Aggregation*, is a general-purpose procedure that reduces the variance of a statistical model. This procedure is especially useful in the context of decision trees. It is known that the averaging of several observations reduces the variance [8, 11, 13, 14, 19, 23]. Thus, the natural way to reduce the variance and, as a result, increase the accuracy of forecasts of a method of statistical training is to take a large number of training samples from the general population, build a forecast model for each training sample and average the forecasts.

Thus, the *Bagging Method* involves creating a large number of copies of the original data set using *bootstrap*, building a separate decision tree for each of these copies, and then combining all the trees to create a single predictive model. It is important to note that each tree is built according to a certain bootstrap sample independently of other trees. Recall that bagging is a special case of the random forest method at $m = p$, when the number of variables in the samples is equal to the number of variables in the initial data set.

In the bagging model, the MSE (*Mean Squared Error*) is 11.97, which is slightly worse than in the training sample, but 2.5 better than the error obtained for the pruned tree. The predicted values of the net hourly power output on the test data set using the bagging model are shown in Fig. 9.

In order to improve the quality of the model, the number of trees created in the bagging model using the “random forest” function is changed. But the correction in the number of trees worsened the MSE on the control data to 12.50 compared to 11.97 in the previous model.

The construction of the random forest model is similar to the bagging model. However, the value of the argument responsible for the number of model predictors required to create each tree node is set to $mtry = 4$.

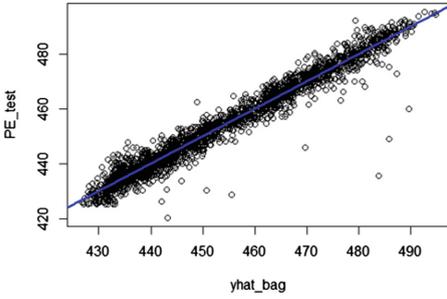


Fig. 9. Estimated values of net hourly power output on the test data set using the bagging model

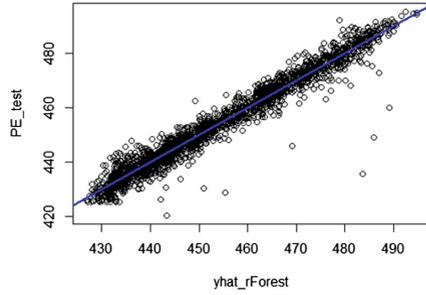


Fig. 10. Estimated values of net hourly electricity output on the test dataset using the random forest model

The values of MSE in the control sample were 11.9, i.e. a random forest led to an improvement in the accuracy of predictions compared to bagging (11.97). The predicted values of net hourly electricity output on the test data set using the random forest model are shown in Fig. 10.

On Fig. 11 shows the calculated importance scores for each model variable. The first one is based on the average decrease in the accuracy of predictions on the remaining (“out-of-bag”) data when the corresponding variable is excluded from the model. The second indicator is a measure of the average increase in the purity of the tree node (“node purity”) as a result of splitting the data on the corresponding variable. For regression, the purity of a node is expressed using RSS.

The results show that in the trees that form the forest, the most important variables are *AT* (temperature) and *V* (the amount of vacuum in the exhaust).

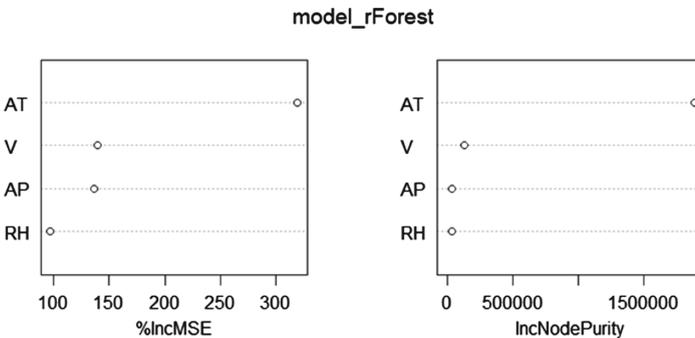


Fig. 11. Graph of indicators of importance of variables of the *Combined Cycle Power Plant* dataset

Boosting. Like bugging, boosting is a common approach that can be applied to many statistical regression and classification methods. We consider boosting in the context of decision trees. The boosting algorithm is similar to the boosting algorithm, but the trees are built sequentially: each tree is grown using information about previously grown trees. Bootstrap samples are not created during boosting, but each tree is built on a modified source data set. Unlike building a large decision tree, which is a narrow approximation of data and can potentially lead to retraining, boosting is *slow learning*.

The graph of the relative importance of variables and the corresponding calculated values of the importance indicator are presented below (Fig. 12).

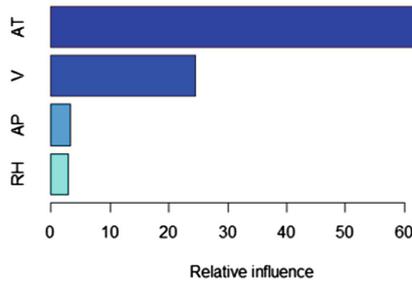


Fig. 12. Graph the relative importance of the variables of the *Combined Cycle Power Plant* dataset

Visually, the histogram shows that AT and V are significantly more important than all other variables. We also provide graphs of the private dependency for these two variables. Graphs of this type illustrate the private effects of certain response variables after fixing the effects of other variables to a constant level (Fig. 13).

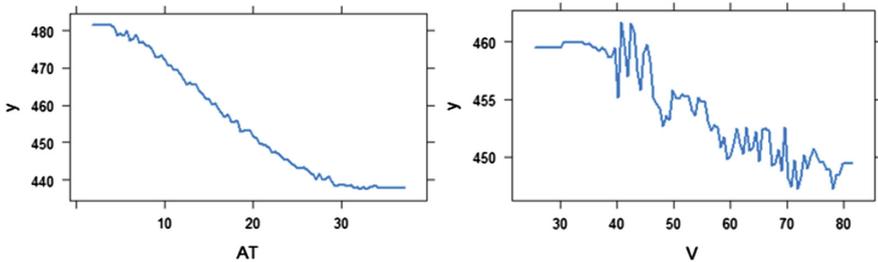


Fig. 13. Private dependency graphs for *AT* and *V* variables of the *Combined Cycle Power Plant* dataset

The graphs show that the value of the net hourly output of electricity is expected to decrease with increasing *AT*. In the case of an increase in *V*, there is

a general tendency to reduce the net hourly output of electricity. The developed boosting model is used to calculate forecasts based on control data and their quality indicators.

The obtained value of MSE in the control sample is 11.02; this value is less than in the random forest (11.90) and much less than in the model built by the method of bagging (11.97). The default in the boost method is the compression parameter $\lambda = 0.001$, but it is easy to change. The new model is built with the compression parameter $\lambda = 0.2$. For the new model, the MSE value is slightly higher in the control sample than in the previous model, at $\lambda = 0.001$ ($11.57 > 11.02$).

The choice of the best model based on trees is based on a comparison of forecast quality indicators (Table 2). The boosting model is the best in terms of quality indicators.

Table 2. Comparative table of forecast quality indicators

Model type	Mean absolute error, MAE	Mean squared error, MSE	Root mean squared error, RMSE	Correlation, cor
Regression tree	4,20	29,3	5,41	0,95
Bagging	2,36	11,97	3,46	0,98
Random forest	2,35	11,90	3,45	0,98
Boosting	2,28	11,02	3,32	0,98

3.3 Implementation of Neural Network Models

Neural networks are one of the most powerful approaches to approximating a function, but it involves problems such as retraining and difficulty of interpretation. The neural network architecture based on multilayer perceptron (MLP) direct propagation was chosen for the implementation of prognostic models. MLP consists of an input layer, an output layer and several hidden layers [14, 17, 19, 21].

Stage 1. Data Collection. The same dataset *Combined Cycle Power Plant* is used to create a prognostic model using neural networks [1].

Stage 2. Research and Data Preparation. Preliminary analysis after loading the data set was performed during simulation using other models.

Neural networks work best when the input data is scaled to a narrow range centered around 0. A summary statistic report shows that the values of the variables range from 0 to 1000.

Typically, the solution to this problem is to scale the data using the normalization or standardization function. If the data distribution corresponds to the normal distribution, then standardization should be used. If the distribution of data is close to uniform or very different from normal, it may be more appropriate to normalize the range from 0 to 1.

$$\tilde{x}_{ik} = \frac{x_{ik} - x_{\min_i}}{x_{\max_i} - x_{\min_i}}; \quad y_{jk} = y_{\min_j} + \tilde{y}_{jk}(y_{\max_j} - y_{\min_j}),$$

where $x_{ik}, y_{jk} - i$ are the input and j output values of the k example of the original sample in traditional units; $\tilde{x}_{ik}, \tilde{y}_{jk}$ are the corresponding to them normalized input and output values; N is the number of examples of the training sample.

$$x_{\min_i} = \min_{k=1, N} x_{ik}; \quad x_{\max_i} = \max_{k=1, N} x_{ik};$$

$$y_{\min_j} = \min_{k=1, N} y_{jk}; \quad y_{\max_j} = \max_{k=1, N} y_{jk}.$$

Stage 3. Learning the Model on the Data. Before learning the model, the normalized data set was divided into a training set (75%) and a test set (25%). The training data set will be used to build a neural network, and the test data set will be used to assess how well the model is generalized for future results.

To model the relationship between the characteristics and the resulting value of the net hourly output of electricity, a multilayer neural network of direct propagation was built. The simplest multilevel network with parameters having only one neuron in one hidden layer is chosen as the first alternative neural network model. Figure 14 presents the topology of the obtained network and the simulation results.

In this simple model, there is one input node for each of the four features, followed by one hidden layer with one neuron and one output node, which predicts the net hourly output of electricity. The diagram also shows the weights for each connection and the offset value indicated for the nodes marked with the number 1. The offset value is a numerical constant that allows you to shift the values in the specified node up or down, approximately as a shift to the linear equation.

The bottom of Fig. 14 shows the number of learning steps and the magnitude of the error – *Sum of Squared Error* (SSE), which is the sum of the squares of the difference between the predicted and actual values. The smaller the SSE, the more accurately the model corresponds to the training data.

Stage 4. Determining the Effectiveness of the Model. Using the neural network model, predictions are generated on the test data set. Since the problem of numerical forecasting is solved, the correlation between the predicted and the actual value of the net hourly output of electricity is calculated to verify the accuracy of the model. If the predicted and actual values are strongly correlated, the model will be useful for determining the net hourly output of electricity. The correlation value is calculated. The obtained correlation value is quite close to 1, which indicates a strong linear relationship between the two variables. This means that the model works quite well even with a single hidden node. However, the result obtained on the basis of the network with the simplest topology needs to be improved.

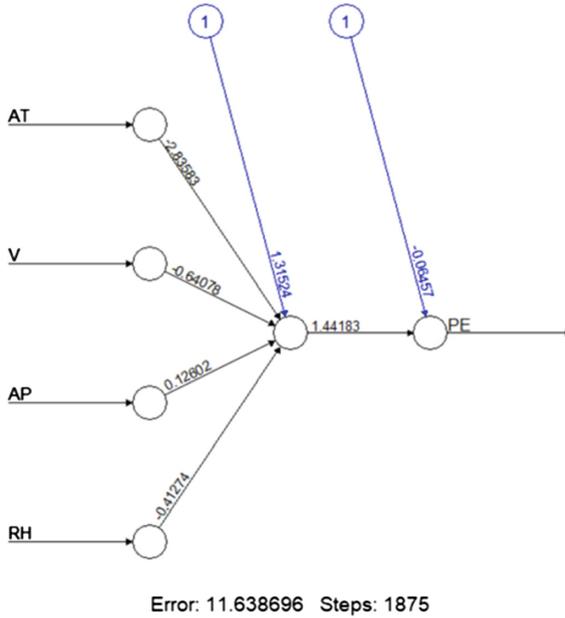


Fig. 14. Visualization of the topology of a simple multilevel network of direct propagation

Stage 5. Improving the Efficiency of the Model. Several more steps have been taken to increase the efficiency of the neural network prognostic model. In particular, additional hidden layers are introduced and the activation function is changed. By making these changes, the foundations are laid for building the simplest deep neural network. The choice of the activation function is made experimentally.

In the neural network, the number of neurons in the hidden layer has increased to five. The presented scheme of network topology (Fig. 15) shows a sharp increase in the number of connections. However, the resulting error (again measured as SSE) decreased from 11.64 of the previous models to 10.66. In addition, the number of stages of training increased from 1875 to 25829. The more complex the network, the more iterations are needed to find the optimal weights.

As an alternative to the sigmoidal activation function used in many libraries, the *Rectified Linear Unit* (ReLU) function was chosen [14]. To solve the problem, the smoothed approximation ReLU, *SmoothReLU*, is used – an activation function defined as $\log(e^x)$.

The next alternative model is a neural network model with two hidden layers, three neurons in each layer, each of which uses the activation function SmoothReLU (Fig. 16).

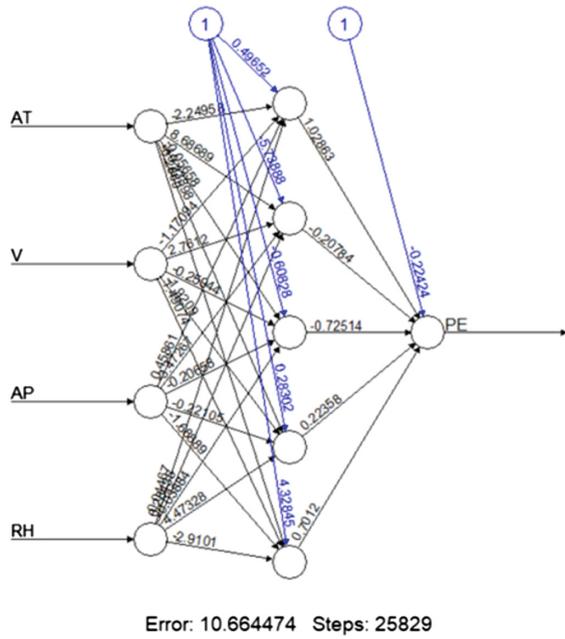


Fig. 15. Visualization of network topology with increased number of neurons in the hidden layer

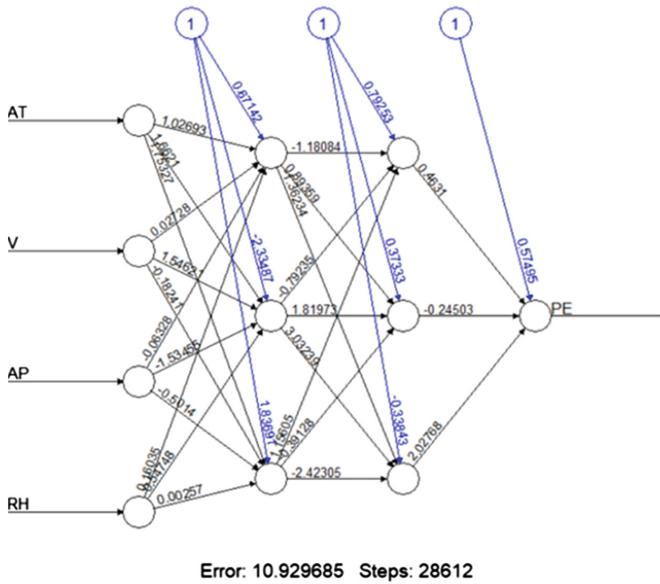


Fig. 16. Visualization of network topology with increasing number of hidden layers

The correlation between the predicted and the actual value of the net hourly power output is slightly worse on the test data set than in the case of previous neural network topologies. This indicates a retraining of the model.

Because the data were normalized before learning the model, the predictions are also in the normalized range from 0 to 1. After applying denormalization to the forecasts, it is obvious that the scale of the new forecasts is similar to the initial values of net hourly electricity output. This allows you to determine the value of the absolute error. In addition, the correlation between non-normalized and output values of net hourly electricity output remains unchanged (0.967). For the test data set, a table was obtained comparing the initial values of net hourly power output, predicted normalized and non-normalized, as well as the error for each observation.

For further comparison with other models, the quality indicators of the neural network model are calculated. The obtained values of quality indicators are quite large, which indicates the low quality of the forecast.

Table 3 shows the quality indicators of alternative prognostic models based on the topology of a multilayer neural network of direct propagation. The M_5 model is the best.

Table 3. Comparative table of alternative prognostic models based on a multilayer neural network of direct propagation

Neural network topology	Number of hidden layers	The number of neurons in the hidden layers	Activation function	Sum of squared error, SSE	Steps	Correlation, cor
M_1	1	1	Sigmoidal	11,64	1875	0,965
M_2	1	2	Sigmoidal	11,18	14419	0,966
M_3	1	3	Sigmoidal	11,05	6919	0,967
M_4	1	4	Sigmoidal	10,66	81084	0,968
M_5	1	5	Sigmoidal	10,66	25829	0,967
M_6	2	1,2	Sigmoidal	11,64	11505	0,965
M_7	2	1,2	SmoothReLU	11,55	13727	0,966
M_8	2	2,2	Sigmoidal	10,85	26010	0,967
M_9	2	2,2	SmoothReLU	10,85	84353	0,967
M_{10}	2	2,3	Sigmoidal	10,85	13977	0,967
M_{11}	2	2,3	SmoothReLU	10,85	16114	0,967
M_{12}	2	3,3	Sigmoidal	10,86	42633	0,967
M_{13}	2	3,3	SmoothReLU	10,93	28612	0,967

Applying neural network models to solve the problem of forecasting, it was found that the topology of neural networks is complicated because it is used for more complex learning tasks. To further increase efficiency, it is necessary to solve

the following problem: when complicating the topology of networks, the value of the gradient increases rapidly when the backpropagation algorithm does not find a solution, as it does not converge over time. To solve the problem, it is necessary to change the number of hidden nodes, apply various activation functions, such as ReLU, adjust the learning speed. This approach to the use of neural networks requires increased network processing time and increased computing power.

3.4 Discussion

The forecast quality indicators of the best prognostic models were analyzed on the basis of three groups of models. The quality indicators of the best models from each group of models are summarized in Table 4. Based on the analysis of the best quality indicators for predicting the value of net hourly electricity output, a boost model with values of mean absolute error $MAE = 2.28$ and square root of root mean square error $RMSE = 3.32$.

Table 4. Comparative table of forecast quality indicators based on three groups of models

Model type	Mean absolute error, MAE	Mean squared error, MSE	Root mean squared error, RMSE	Correlation, cor
Improved regression model	3,61	21,39	4,62	0,962
Boosting model	2,28	11,02	3,32	0,980
Neural network model	4,66	27,67	5,26	0,967

Compared to previous studies on the same data set [12,24], the application of the developed methodology improves the quality of the model on the best of the alternative models: MAE by 23% and more than 14% RMSE. This indicates the effectiveness of the proposed methodology.

4 Conclusions

The article discusses the forecasting methodology based on machine learning methods. The methodology consists of five stages: the stage of data collection, the stage of research and data preparation, the stage of training the model on data, the stage of determining the effectiveness of the model, the stage of improving the efficiency of the model. An additional visualization procedure accompanies each of the stages of the methodology. The methodology is designed to solve

forecasting problems based on the use of various methods, models and their combinations. Examples of using the methodology for solving the problem of forecasting with the help of regression models that take into account the peculiarities of the relationship of initial variables are considered; models based on decision trees (basic models and their ensembles); models based on multilayer neural networks of direct distribution. The base data set was a set for predicting the power of a combined cycle plant. The effectiveness of each of the alternative models belonging to three groups is analyzed. The main feature of the methodology is to identify and study the effectiveness of machine learning models and their further improvement. Various quality indicators were used to assess the quality of the models. The results obtained indicate good quality of the predicted values with the improved regression model, the bagging model, the boosting model, and the best neural network model. When implementing regression models, a number of assumptions were made: normality, independence, linearity, and homoscedasticity. These assumptions were tested using tests. Diagnostic methods were applied to each of the alternative models. The results of applying the methodology were compared with the results published earlier. Using the proposed methodology can significantly improve the forecasting process. As a future work on this problem, other learning models such as k-NNs, support vector machines, other neural network topologies, and combinations of these can be used to enhance the ability to compare prediction accuracy and learning performance.

References

1. Machine learning repository, combined cycle power plant data set. <https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant#>
2. Allison, P.: *Multiple Regression: A Primer*, p. 224. Pine Forge Press, Thousand Oaks, CA (1999)
3. Bari, A., Chaouchi, M., Yung, T.: *Predictive Analytics For Dummies*, p. 435. John Wiley and Sons, Inc (2016)
4. Bidyuk, P., Gozhyj, A., Kalinina, I., Vysotska, V., Vasilev, M., Malets, R.: Forecasting nonlinear nonstationary processes in machine learning task. In: *Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP 2020*, pp. 28–32, no. 9204077 (2020)
5. Bidyuk, P., Gozhyj, A., Matsuki, Y., Kuznetsova, N., Kalinina, I.: *Advances in intelligent systems and computing. Lect. Notes Data Eng. Commun. Technol.* **1246**, 395–408 (2021)
6. Bidyuk, P., Kalinina, I., Gozhyj, A.: An approach to identifying and filling data gaps in machine learning procedures. *Lect. Notes Data Eng. Commun. Technol.* **77**, 164–176 (2021)
7. Bojer, C.S.: Understanding machine learning-based forecasting methods: a decomposition framework and research opportunities. *Int. J. Forecast.* **38**(3), 7 (2022). <https://doi.org/10.1016/j.ijforecast.2021.11.003>
8. Breiman, L., Friedman, J., Stone, C., Olshen, R.: *Classification and Regression Trees: Clusterization and Recognition*, p. 368. Chapman and Hall, CRC Press (1984). <https://doi.org/10.1201/9781315139470>

9. Dong, H., Gao, Y., Fang, Y., Liu, M., Kong, Y.: The short-term load forecasting for special days based on bagged regression trees in Gingdao, china. *Comput. Intell. Neurosci.* **2021**, 16 (2021). <https://doi.org/10.1155/2021/3693294>
10. Hartomo, K., Nataliani, Y.: A new model for learning-based forecasting procedure by combining k-means clustering and time series forecasting algorithms. *PeerJ Comput. Sci.* **7**(e534), 29 (2021). <https://doi.org/10.7717/peerj-cs.534>
11. James, G., Witten, D., Hastie, T., Tibshirani, R.: *An Introduction to Statistical Learning*. STS, vol. 103. Springer, New York (2013). <https://doi.org/10.1007/978-1-4614-7138-7>
12. Kaya, H., Tüfekci, P., Gürgen, S.F.: Local and global learning methods for predicting power of a combined gas and steam turbine. In: *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE 2012*, pp. 13–18 (2012)
13. Kelleher, J., Namee, B., D'Arcy, A.: *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*, p. 624. The MIT Press (2015)
14. Lantz, B.: *Machine Learning with R. Expert techniques for predictive modeling*, p. 458. Packt Publishing (2019)
15. Long, J., Teetor, P.: *R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics*, p. 600. O'Reilly Media, Inc (2019)
16. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods. *PLoS ONE* **13**(3), 26 (2018). <https://doi.org/10.1371/journal.pone.0194889>
17. Neapolitan, R., Jiang, X.: *Artificial intelligence with an introduction to Machine Learning*, p. 480. CRC Press, Taylor and Francis Group, Informa business (2018)
18. Nielsen, A.: *Practical Time Series Analysis: Prediction with Statistics and Machine Learning*, p. 505. O'Reilly Media, Inc (2019)
19. Patel, A.: *Hands-On Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data*, p. 562. O'Reilly Media, Inc (2019)
20. Petropoulos, F., Apiletti, D., Assimakopoulos, V.: Forecasting: theory and practice. *Int. J. Forecast.* **38**(3), 705–871 (2022). Elsevier, B.V.: On behalf of Int. Inst. Forecast. (2021). <https://doi.org/10.1016/j.ijforecast.2021.11.001>
21. Rumelhart, D., Hinton, G., Williams, R.: Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986). <https://doi.org/10.1038/323533a0>
22. Sarker, I.H.: *Machine learning: algorithms, real-world applications and research directions*. *SN Comput. Sci.* **2**(3), 1–21 (2021). <https://doi.org/10.1007/s42979-021-00592-x>
23. Scavetta, R., Angelov, B.: *Python and R for the Modern Data Scientist: The Best of Both Worlds*, p. 198. O'Reilly Media, Inc (2021)
24. Tufekci, P.: Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *Int. J. Electr. Power Energy Syst.* **60**, 126–140 (2014)
25. Wickham, H., Grolemund, G.: *R for Data Science. Import, Tidy, Transform, Visualize, and Model Data*, p. 520. O'Reilly Media, Inc. (2017)



The Comprehensive Model of Using In-Depth Consolidated Multimodal Learning to Study Trading Strategies in the Securities Market

Nataliya Boyko^(✉) 

Lviv Polytechnic National University, Lviv, Ukraine
nataliya.i.boyko@lpnu.ua

Abstract. The paper describes the relevance of machine learning methods, namely training with reinforcement, to the problems of predicting financial time series. An overview of existing applications based on machine learning in the issues of financial market forecasting is presented. The reasons for the popularity of my research topic are highlighted both from the scientific (increasing the number of publications on issues relevant to the research topic over the past five years) and from the practical side. The analysis of scientific works, the subject and purpose of which are related to the issues and objectives of my research and their features are presented. The main problems associated with the problem of predicting stochastic time series are identified. According to the analysis, the purpose of work is defined, and also the list of tasks for the achievement of the set goal is made. The article is devoted to studying the use of the ensemble of neuro-learning networks with the strengthening of the securities trading market. The practical significance of the work is to use the model of efficient distribution of investments in the market. This paper will explore a set of models that use in-depth consolidated learning to explore trading strategies to maximize return on investment. The potential of using acting-critical models as an ensemble has been studied. Models such as Proximal Policy Optimizer (PPO), Advantage Actor-Critic (A2C) and Deep Determinist Police Gradient (DDPG) were used to teach trading strategy. To adapt the model to different situations, analyzes are analyzed according to three algorithms: the Dow Jones average and a portfolio that minimizes fluctuations in the Charpy ratio by balancing risk and return. The article compares ensembles by the method of fixing deep neural networks. To optimize the balance of risk and profit, the indicators of the ensemble model are analyzed. The ensemble and three-component models that apply well to market collapse conditions are considered. The models have learned to use the turbulence index for early stock sales to minimize losses during a stock collapse. The turbulence index threshold is demonstrated using ensemble models to regulate risk avoidance.

Keywords: Proximal policy optimization · Actor-critic · Deterministic deep gradient policy · Depth neural network · Dow jones industrial average · Solving markov processes

1 Introduction

Given the global changes, it is difficult for people to rely solely on savings to ensure their financial security. There are two reasons for this: first, free money in your bank account is a loss of opportunity because it does not allow you to make more money, and second, it does not have the potential to overcome inflation. If we consider the inflation index (consumer price index [10]), the inflation rate was 20.5%, which shows the need to change the approach to saving.

Investing and trading (a speculative buying and selling of financial instruments (stocks, bonds, futures, options, currencies, etc.) to make a profit from price changes) are two very different methods of trying to make a profit on financial markets [20]. Both investors and traders seek to make a profit through market participation. In general, investors seek to make more profits over the long term through purchasing and holding financial instruments. On the other hand, traders take advantage of both rising and falling markets to enter and exit positions in a shorter period, earning less but more frequent profits.

Trading involves more frequent transactions, such as buying and selling securities, commodities, currency pairs or other instruments. The goal is to get a profit that exceeds the effectiveness of the "buy and hold" investment [19]. Although investors can be satisfied with an annual return [21] of 10% to 15%, traders can receive 5% or more profit per month. Profit is generated by buying any of the above instruments at a lower price and selling it at a higher price in a relatively short period. The opposite is also possible: trading profits can be obtained by selling any of the above instruments at a higher price and buying it to cover at a lower price (known as "selling short" [24]) to make a profit when the market falls.

The net profit of both investors and traders depends on which of the three actions they decide to do: buying, selling or holding financial instruments. The right actions depend on many factors [15] and dimensions [16] of the financial market. Therefore, determining the right actions requires special knowledge from investors.

According to their research [12–22], economists have proposed specific strategies and guidelines for finding the best option for trading in the stock market. However, the capital of several investors decreased when they tried to trade based on the recommendations of these strategies. This means that the stock market needs more in-depth research to provide more guarantees of success for investors.

An investment strategy is an essential component of trade. Then be a technical analysis of short-term opportunities or fundamental analysis of long-term investments. But even with all the statistics, it is often difficult to find the right strategy in a dynamic and comprehensive market. Most traders do not beat

the market and even go into the red. To do this, you need to train the following agents to acquire an ensemble of strategies: Proximal Policy Optimization (PPO), Critic Actor Advantage (A2C), and Deep Deterministic Policy Gradient (DDPG). The ensemble's designs combine the capabilities of all three algorithms, allowing the system to adapt to different market situations.

The system has been tested on the historical data of 30 shares of the DOW index, which has sufficient liquidity. The effectiveness of the agents and the ensemble is comparable to the strategy of minimal variations: portfolio performance and the Dow Jones Industrial Average (DJIA). The article shows that this approach surpasses all three separate algorithms and the comparative strategy for risk-adjusted income indicators (Sharpe ratio). The work's scientific value lies in the study of the scalability of in-depth learning with reinforcement, with the possibility of application to large-scale problems in combination with noisy and nonlinear patterns in financial time series.

The topic's urgency is more efficient distribution of investments in the securities market and improving hedge funds.

The study aims to create an agent ensemble of more effective models than the model parts of the ensemble and selected for comparison strategies. Achieving this goal involves solving the following tasks:

- Develop learning environment models and a testing ensemble.
- Develop an algorithm for effective integration of models into the ensemble.
- Compare the effectiveness of the selected model with existing strategies and models.

The practical significance of the work is to use the model for the efficient distribution of investments in the market. The research object is the securities market, which studied more successful investment strategies. All work was performed on one local personal computer using the OpenAI service. Research in this work is characterized by solving the current scientific and practical decision-making problem in the management of integrated investment in conditions of risk and uncertainty through methods and tools for investment decision-making in situations of luck - the conducted research allowed to receive individual results.

Thus, in recent years, scientists and proponents of technical analysis have made serious attempts to use various mathematical and statistical models and, more recently, machine learning (ML) methods to try to predict financial markets (financial time series). It is also worth noting that the most popular practices described in the technical literature are supervised learning in the context of a regression problem. The problem is formalized as a training ML model on available training data with its subsequent assessment of how well it summarizes hidden dependencies by testing it on test data.

Therefore, the relevance and interest in using in-depth training with reinforcement to predict financial time series were considered scientifically and practically.

2 Literature Review

Due to the complexity of the market system, most traders cannot predict price movements. This leads to 70% of them not earning money [10]. This means that market inefficiencies are created. For analysis, the securities market often uses time series analysis [20], which leads to suboptimal results, as most of the share price of an individual company depends on fundamental factors such as the income of this company, the state of the economy and the securities market in general, industry analysis, etc. But while traditional time series analysis methods do not provide small market analysis results, machine learning methods, and more specifically deep neural networks, showed results that would be appropriate to study how on.

Modern applications of Depth neural network (DNN) in the market analysis are used discrete and constant states and spaces of action and use one of the following approaches:

- The method with only the critic.
- The technique with only the actor.
- The method actor-critic.

Models with a constant space of action provide better control capabilities for the agent than models with discrete space [22]. It has been shown that models of the actor-critic method, such as PPO [19], A2C [21], and DDPG [24], demonstrate high accuracy in the analysis of the securities market papers [15].

The main idea of the actor-critic approach is to update the actor-network, which offends strategy, and network criticism, which reflects the value function. Critic roughly estimates the value of the value function. At the same time, the actor updates the distribution of probabilities of the strategy of choosing one or another action, directed by the critic with its gradient improvement principles. Over time, the actor learns to accept better solutions, and the critic evaluates these solutions better.

The actor-critic approach proved its ability to adapt to complex environments and used states to play popular video games, such as Doom [1].

Thus, the actor-critic approach is promising for research in trading with an extensive portfolio of shares. One of the problems with the system for learning and using models is that different market situation, and strategies are needed, which in connection with some models cannot be generalized by this structure. Combining models in the ensemble allows the models to improve the results significantly. They combine their advantages and compensate for the disadvantages, which leads to improvement and generalization of learning problems [16].

As the problem is relevant, numerous studies have been conducted on reinforced learning and its application to financial time series. Article [23] presents a program based on reinforced knowledge. From a typical approach to learning from mistakes, the decision is transformed into a learning method focused on compliance with the goal. The possibility of real-time financial market forecasting and traditional eligibility criteria were also considered. It is shown that the

efficiency of systems based on “actor-critic” algorithms is superior to other studied alternatives. At the same time, the proposed methods based on “actor-only” algorithms have also shown good efficiency.

Thus, as investment and trade become more popular, research into the automation of human interaction with financial markets is of great interest to scholars and professional investors, both scientifically and practically. We can see that using machine learning techniques such as reinforced learning and exceptionally in-depth learning with reinforcement to predict financial time series shows promising results. It can potentially replace a person in interaction with financial markets. It is worth noting that most of the analyzed studies aim to find a way to apply known algorithms in economic time series and build ensembles, which will provide more guarantees of success to investors to increase net income in real-time.

3 Materials and Methods

3.1 Market Indicators

Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follow a table, figure, equation etc., do not have an indent [18,22].

Subsequent paragraphs, however, are indented.

The moving average is one of the tools for analyzing random processes and time series, which calculates the average subset of values. The size of the subset of the calculated mean can be both constant and variable. The moving average may have weights (1), for example, to increase the impact of newer data compared to old ones.

$$y_t = \sum_{r=-q}^{+s} a_r x_{t+r}, \quad (1)$$

where x_t is the time series, y_t is the moving average, a_r is the weight of the value of the series, $-q$ is the beginning period for which the moving average is calculated, $+s$ is the end of the period.

The Moving Average/Divergence Indicator (MADI) is the following on the trend of the dynamic indicator. It shows the relationship between the two moving average prices. The Technical Indicator is constructed as a difference between two exponential moving averages (EMA) with periods of 12 and 26 (2).

$$\text{MADI} = \text{EMA}(\text{close}, 12) - \text{EMA}(\text{close}, 26), \quad (2)$$

where EMA is the exponential moving average, close is the closing price market [1,8].

The Relative Strength Index (RSI) is an indicator of technical analysis that measures the strength of the trend and the probability of its change. To calculate

the RSI, positive (U) and negative (D) price changes are used. The day is called “growing” soft closing price is higher than yesterday (3).

$$\begin{aligned} U &= \text{close}_{\text{today}} - \text{close}_{\text{yesterday}}, \\ D &= 0 \end{aligned} \tag{3}$$

The day is called “descending” if the closing price is lower than yesterday (4).

$$\begin{aligned} D &= \text{close}_{\text{yesterday}} - \text{close}_{\text{today}}, \\ U &= 0 \end{aligned} \tag{4}$$

If the closing prices yesterday and today are equal, then U and D are equal to 0. After the values of U and D are smoothed using the moving average with the period N . Thus, the “relative force” (RS) is calculated (5).

$$\text{RS} = \frac{\text{EMA}(\text{close}, N) \text{ of } U}{\text{EMA}(\text{close}, N) \text{ of } D} \tag{5}$$

Based on RS, IBC is calculated (6) [8]:

$$\text{IBC} = 100 - \frac{100}{1 + \text{RS}} \tag{6}$$

The Commodity Channel Index (CCI) is a technical indicator based on analysing the current price change from the average value for some period. One hypothesis is that if the price is shifted from its moving average by an interval greater than expected for the period under consideration, a change occurs trend [12, 26]. To calculate the index is necessary to determine a typical price. Typical, the cost is calculated as follows (7):

$$p_t = \frac{\text{high}_t - \text{low}_t + \text{close}_t}{3}, \tag{7}$$

where, p_t is the typical price, high_t is the maximum price for the period t , low_t is the minimum price, close_t - closing price.

The index itself is calculated as follows (8)–(9) [8, 17]:

$$\text{CCI}_t = \frac{p_t - \text{SMA}_n(p_t)}{\text{MAD}_n(p_t)}, \tag{8}$$

where CCI_t is the Commodity Channel Index for the study period t , $\text{SMA}_n(p_t)$ is the simple moving average of a typical price, $\text{MAD}_n(p_t)$ is the absolute average deviation of the typical price from their moving average for the period n .

$$\begin{aligned} \text{SMA}_n(p_t) &= \frac{1}{n} \sum_{i=0}^{n-1} p_{t-i}, \\ \text{MAD}_n(p_t) &= \frac{1}{n} \sum_{i=0}^{n-1} |p_{t-i} - \text{SMA}_n(p_n)| \end{aligned} \tag{9}$$

The Medium Direction Index (MDI) is a technical indicator that signals that the market is in a solid and unchanging trend. For calculation of the index, it is necessary to determine the index of the direction of the hand (10)–(13):

$$\begin{aligned} +M_t &= \text{high}_t - \text{high}_{t-1}, \\ -M_t &= \text{low}_{t-1} - \text{low}_t, \end{aligned} \quad (10)$$

where $+M_t$ is the positive movement, $-M_t$ is the negative movement, high_t , low_t is the current maximum and minimum market prices, high_{t-1} , low_{t-1} are the maximum and minimum market prices for the previous period.

$$\begin{aligned} +DM_t &= \begin{cases} +M_t, & \text{if } +M_t > -M_t \text{ and } +M_t > 0, \\ 0, & \text{if } +M_t < -M_t \text{ or } +M_t < 0 \end{cases}, \\ -DM_t &= \begin{cases} 0, & \text{if } -M_t < +M_t \text{ or } -M_t < 0, \\ -M_t, & \text{if } -M_t > +M_t \text{ and } -M_t > 0 \end{cases}, \end{aligned} \quad (11)$$

where $+DM_t$, $-DM_t$ are the positive and negative directional movements.

$$\text{DXI}_t = 100 \cdot \frac{|+DM_t - -DM_t|}{+DM_t + -DM_t}, \quad (12)$$

where DXI_t is the direction index.

$$\text{ADI}_t = \text{MA}(\text{DXI}_t, n), \quad (13)$$

where ADI_t is the Average Direction Index, MA is the moving average with period n .

3.2 Market Model

In this work, the stock market is modelled as the Markov process solution (MPS) [4, 26], as the agent's task was chosen to maximize the expected income.

To model the stochastic nature of the dynamic Securities Market, MPS used becomes as follows [11, 14, 17]:

- Status $s = [p, h, b]$: a vector comprising share price $p \in R_+^D$, quantity shares in the portfolio $h \in Z_+^D$, the final balance in money $b \in R_+^D$, where D – dimension of space, number of shares 30, $+$ in the notation reflects non-negative value.
- Action a : vector D elements. Actions on each of the shares include: sales, purchase, retention, which results in a decrease, increase, and no changes in h , respectively.
- Function reward $r(s, a, s')$: straight award from taking action a in a state s and the transition to the state s' .
- The strategy $\pi(s)$: trading strategy for the state s , which is a division probability for actions that are possible in state s .

- Q-value $Q_\pi(s, a)$: the expected decision to award a underway s according to the strategy π .

These assumptions are used to improve the model’s practicality: transaction cost, market liquidity, risk aversion, etc. [3, 18].

- Market liquidity: the stock is executed at the market price at that moment. We believe that our agent does not significantly affect the Securities Market.
- An integral balance $b \geq 0$: possible actions cannot lead to a negative money balance.
- The cost of the transaction, each transaction has Commission. There are many different you-commissions, such as - commissions for exchange, commissions for execution, and commissions for provision of services. Other brokers have various commissions. Despite additional diversity, we believe that all commissions in the model will be equal to wool 0.1% of the value of every transaction (14):

$$c_t = p^t k_t \cdot 0, 1\%, \tag{14}$$

where k_t is the number of purchased active of t -company.

- Aversion to risk in the market’s collapse [7, 17] is some unpredictability circumstances that may provoke a market collapse, such as wars, the destruction of market bubbles, domestic debt, and financial crises. To control risk metrics, we use turbulence, which measures abrupt changes in the value of the asset (15):

$$\text{turbulence}_t = (y_t - \mu) \sum_{?}^{-1} (y_t = \mu)' \in \mathbb{R}, \tag{15}$$

where $y_t \in \mathbb{R}^D$ is the profit from the sale of shares for the period t , $\mu \in \mathbb{R}^D$ are the average historical profits, $\sum \in \mathbb{R}^{(D \times D)}$ is the covariation of historical income. When turbulence is higher than the threshold, which means the extreme state of the market, we prohibit any purchase or sale, and our agent is forced to sell all shares.

3.3 Environment

Before training the neural network of deep learning with fixed, It is necessary to create an environment for simulating trading in the re-world carefully. This environment should be interactive and allow agents to study. It is essential to consider various information for trading, such as historical stock prices, stock controls by other agents, technical indicators, etc. The model agent needs to get all this information through an environment and take the steps outlined in the previous paragraph (Market Model). To create an environment and train the agent, was used OpenAI gym [2, 5, 11, 23].

- Space condition: used vector size 181, which is given in seven parts, which reflect the space of market states with some number of shares (30): [bt, pt, ht, Mt, Rt, Ct, Xt]. Where:

- $b_t \in \mathbb{R}_+$: available money balance at time t .
 - $p_t \in \mathbb{R}_+^{30}$: share price at the time of market closing on day t .
 - $x_t \in \mathbb{Z}_+^{30}$: the number of shares owned by the agent.
 - $M_t \in \mathbb{R}^{30}$: Technical Convergence/Divergence Indicator Sliding (TCDIS) at the time of market closure. TCDIS - is one of the most popular change indicators in averages [8].
 - $R_t \in \mathbb{R}_+^{30}$: Relative Strength Index (RSI) is calculated at the cover price. RSI measures how much prices have changed. If these are on the supported support level, the stock is too much sell, and we can use the action - buy. If the price is kept - there is a level of resistance, which means that the stock is bought too much, and we can use the action - sale. [8].
 - $C_t \in \mathbb{R}_+^{30}$: Commodity Channel Index (CCI) is calculated using the highest, lowest, and closing prices. CCI compares the current price with the average cost for a certain period to help choose between buying and selling [12].
 - $X_t \in \mathbb{R}^{30}$: The Average Direction Index (ADI) is calculated by the highest, lowest, and closing prices. ADI measures how much the price is moving [4, 13].
- Space Action: For every shared space with action $-m, \dots, -1, 0, 1, \dots, n$, where n and $-m$ are the number of shares that can be bought and sold, respectively. Also, on n and m is the condition, $n, m \leq hmax$, where $hmax$ is the maximum number of shares per transaction. Thus, the size of the state space becomes $(2k + 1)30$. Then the state space is normalized to $[-1, 1]$; yes, both the A2C and PPO algorithms use the Gaussian distribution for their strategy function.

Organization of computer memory: this amount shares types of multimodal data, attributes, neural network layers, and cost batch size significant memory [4, 8, 9, 12, 13]. To solve the problem of memory needs, use a boot-on-demand method for efficient memory use.

As shown in Fig. 1, this method does not store anything in memory but generates as needed. Memory is used only when prompted, which reduces use.

3.4 Sampling Rate

Highly liquid stocks of the DOW index were chosen for training, as the environment considered in this paper did not simulate liquidation delays financial assets. Index shares were selected for analysis because Index funds are a reliable investment strategy that is currently the best of 99% of traders. Their approach [7, 10, 25] analyzed the efficiency algorithm.

Data can be obtained from the Composter database using Wharton Research Data Services (WRDS) [3, 6]. The dataset is divided into two samples: training and test. Training and validation were conducted on the training sample.

Three-component models were trained during the training stage, namely PPO, A2C, and DDPG. After that, the validation stage was necessary to adjust and select hyperparameters of models, such as learning rate and quantity of

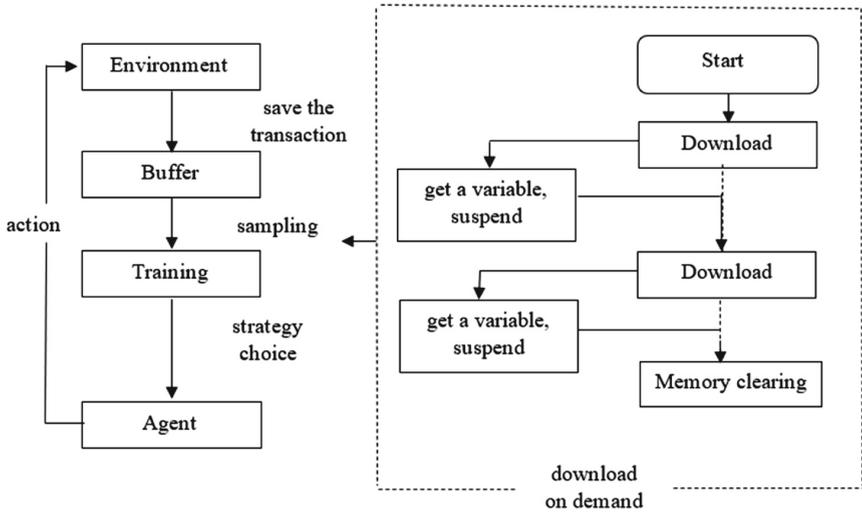


Fig. 1. Schematic representation of the download-on-demand method.

learning episodes. After that, we collected information on a test sample on the behaviour of all models and test strategies for comparison.

The training sample included data from 01.01.2009 to 01.10.2015. The validation sample included data from 01.10.2015 to 01.01.2016. The training sample consists of the period from 01.01.2016 to 08.05.2020.

The model continues to learn even during the testing stage to improve the model efficiency. This helps the agent to adapt to the dynamic market.

3.5 Depth Neural Network

Deep learning is a field of machine learning based on algorithms that try to model high-level abstractions in the data, applying a depth graph with several machining layers constructed of several linear or nonlinear transformations.

In-depth learning is part of a broader family of machine learning methods - based on learning data characteristics. Observations (e.g. an image) can be represented in many ways, such as vector brightness values for pixels, or in a more abstract way as set edges, areas of a specific shape, etc. Some representations are better than others. In simplifying the learning task (for example, facial recognition or expression) face). One of the promises of in-depth learning is to replace the signs of manual labour with practical algorithms for automatic or semi-automatic learning of characters and the hierarchical selection of features.

Research in this area tries to make better representations and create models to teach these representations from large-scale unlabeled data. Some of these representations were inspired by the achievements of neuroscience and the motives of schemes for processing and transmitting information in the nervous system, such

as neural coding, which tries to determine the connection between various stimuli and associated neural responses in the brain. In-depth learning algorithms are based on distributed representations. The assumption underlying distributed models is that the observed data are generated by the interaction of factors organized in it.

In-depth learning assumes that these factors correspond to different levels of abstraction or construction. To ensure different degrees of Variable amounts and sizes of layers can be used. Deep learning uses this idea of hierarchical explanatory fa- where lower-level concepts are taught in more abstract ways. These architectures are often built-in layers greedy method. Deep understanding allows you to unravel these abstractions and pick out good signs for education. For guided learning tasks, in-depth learning methods are avoided by designing features by converting data into compact, intermediate views like the main components and removing the layered structures that eliminate redundancy in presentation. Many in-depth learning algorithms are applied to spontaneous training. This is an essential advantage because unlabeled data is usually thicker than labelled. Examples of deep structures that can be trained spontaneously are neural compressors of history and deep networks of beliefs. A deep neural network is an artificial neural network with several hidden layers. Like conventional neural networks, DNNs can model complex nonlinear relations between elements. In the learning process, The DNN model obtained tries to represent the object as a combination of simple primitives (for example, in the problem of recognizing faces by such primitive you can be parts of the face: nose, eyes, mouth, and so on) [14,18].

Additional layers allow the construction of abstractions of ever-higher levels, which enables the construction of models for recognizing complex real-world objects. As a rule, deep networks are built as networks of direct distribution. However, recent studies have shown the technique of depth- training for recurrent neural networks. Convolutional neural networks are used in machine vision, where this approach has proven effective. Active. Convolutional neural networks have also been used for recognition languages. Learning deep neural networks can be done with the help of the usual error backpropagation algorithm. There is an extensive number of modifications to this algorithm. Thus, several can be used rules of adjustment of scales. For example, the training of weights $\omega_{ij}(t)$ algorithm stochastic gradient descent rhythm (16):

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \nu \frac{\delta C}{\delta \omega_{ij}}. \quad (16)$$

In the deep neural network scheme (see Fig. 2), 1 is the input signals, 2 is the scales connections between neurons, 3 is the output signals, and 4 is artificial neurons.

3.6 Q-learning

Q-learning is an algorithm of modeless learning with reinforcement. Q-learning aims to create strategy π , which tells the agent to go for action in certain cir-

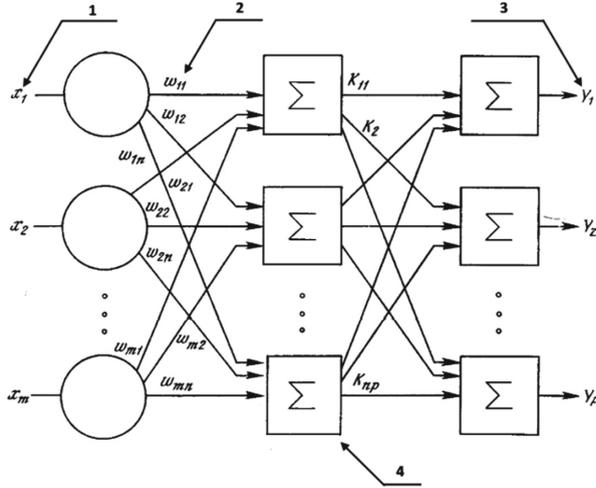


Fig. 2. Scheme of a deep neural network.

cumstances. It does not require a model environment (hence the refinement “modeless”) and can solve stochastic transitions and rewards without adaptations.

For any completed MPS, Q-learning finds an optimal strategy. It maximizes the expected value of the total gardens over any successive steps starting with the current state.

The weight for the step from the state to δ_t steps in the future is calculated as μ^{δ_t} , where μ is the depreciation coefficient e is a number from 0 to 1 ($0 \leq \mu \leq 1$), and gives the effect of prioritizing the evaluation of previously received rewards above those received later [7, 23].

The core of the algorithm is the Bellman equation, which uses a weighted day of the old value and new information (17):

$$Q^{new}(s_t, a_t) = Q(s_t, a_t) + \alpha \cdot (r_t + \mu \cdot \max Q(s_{t+1}, a) - Q(s_t, a_t)), \quad (17)$$

where r_t is the reward obtained in the transition from state s_t to state s_{t+1} , and α is the pace of learning.

The learning rate, or step size (α), determines how much the new information will override the old one. A factor of 0 will make the agent not learn anything (using only a priori knowledge). In contrast, coefficient 1 will make the agent consider only the latest information (ignoring prior knowledge to explore opportunities). In an entirely de- in terminated environments, the optimal learning rate is $\alpha_t(s, a) = 1$. When giving is stochastic, the algorithm will still match under some technical conditions at a pace of learning that requires it to be reduced to zero. In practice, often use a constant rate of knowledge, such as, for example, $\alpha_t = 0.1$ for all t .

3.7 Reinforcement Learning

Reinforcement learning includes an agent, a plurality of S states, and a plurality of Action A . Acting $a \in A$, the agent moves from one state to another. You-performing a certain action in a certain state provides the agent with a reward. The primary agent's goal is to maximize his reward function. He does this by adding the maximum reward that can be achieved from future states to achieve its current state, thus affecting current action by a potential future reward. This is a possible reward, the weighted sum of the mathematical expectation of the rewards of all subsequent steps, starting from the current state. Mathematically, the task of reinforcement learning is a Markov decision-making process and consists of:

- Set of states' S .
- Set of actions A .
- Reward function $r(s, a)$ – a function that displays related condition and action to reward.

Review of the Markov decision-making process:

- The agent performs an action a_1 in the state s_1 .
- Medium is steam state values and reward action and returns $r_1(a_1, s_1)$.
- Agent later, the following state (s_2).

This is repeated until we reach the last state. Task agent: to find a strategy of π following conditions and actions that maximize an acute measure of amplification. In general, the environment is expected to be deterministic, i.e. the performance of the same activities in the same state in two different cases. It can lead to other states or additional value rewards. However, the medium must be stationary, i.e. probability of transitions to the state or the receipt of specific gain signals will not change over time. Each strategy leads to a particular trajectory of states, actions and rewards. We use the function of the values of V to determine how well specific activities and conditions are for the agent who follows the strategy. The optimal Q-value function Q^* gives us the maximum possible value from a given pair of state actions in any process. One of the fundamental properties is that it must satisfy the optimality equation Bellman. The optimal Q-value for the state-action couple equals the greatest reward that the agent can receive from the action in the current state. The equation has the form (18):

$$Q^*(s, a) = E[R_t + 1 + \mu \cdot \max_{a'} Q^*(s', a')]. \quad (18)$$

Because using the Bellman equation it is possible to calculate each of the steps, it allows us to determine the optimal strategy: we know the value of the next pair of states and actions, so the optimal approach is to reproduce the actions, which maximize the expected value (19):

$$R + \mu Q^*(s', a'), \quad (19)$$

where R is an immediate reward, $Q^*(s', a')$ is the optimal Q-value for the state in which we find ourselves, μ is the depreciation factor. So the optimal strategy π^*

can allow us to choose the best actions for a particular state based on expected reward (see Fig. 3).

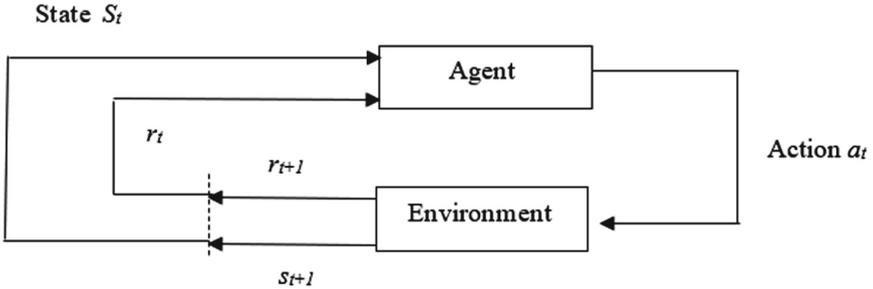


Fig. 3. Schematic representation of the agent-environment interaction.

3.8 Actor-critic Model

Pseudocode of the Actor-Critic algorithm (see Algorithm 1), where a_{it} is the selected action, s_{it} is the state, in which the agent is located, V_{θ}^{π} is the average base value, V_{θ}^J is the average cumulative award, A is the advantage.

Algorithm 1: Actor-Critic algorithm

Repeat:

1. : Generate trajectories $\{(s_i, a_i)\}$ using $\pi_{\theta}(a|s)$
 2. : Update weights $\hat{V}_{\theta}^{\pi}(s)$ (1 method)
 3. : $\hat{A}^{\pi}(s_i, a_i) = R(s_i, a_i) + \hat{V}^{\pi}(s_{i+1}) - \hat{V}^{\pi}(s_i)$
 4. : $\nabla_{\theta} J(\theta) \approx \sum_i \nabla_{\theta} \log(\pi_{\theta})(s_i|a_i) \hat{A}^{\pi}(s_i, a_i)$
 5. : $\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta)$
-

Actor-Critic comes from a pair of models that approximate the function cost and strategy. The network that brings the strategy closer is called the Actor, and the network a cut approximates the value function is called a critic.

At the beginning of training, the actor does not know which action to choose, so he chooses an action randomly. The critic observes the actor’s action and gives feedback. Study-Following this response, the actor updates the strategy, while the critic updates the function cost.

3.9 Advantage Actor Critic (A2C)

A2C [21] is a standard algorithm of the actor-critic model, and it is used as a component of the ensemble model. A2C is used to improve the grad-change strategy. A2C uses “advantage” to reduce variability gradient strategy. Instead of just predicting the reward function, the network the critic assumes the position of superiority. Thus, the assessment of actions depends on how good the action

is and how better it could be. This reduces the high variability of the strategy but makes the model less small and adaptive to abrupt changes.

A2C uses a copy of the same agent to update the gradient. That of one sample. Each agent independently interacts with one environment. After the agent completes the gradient calculation on each iteration, A2C is used. It uses a coordinator to average the gradient between all network agents.

Thus, the global network updates the actor and the critic.

Goal function for A2C algorithm (20):

$$\nabla J_{\theta}(\theta) = E \sum_{t=1}^T \nabla_{\theta} \log(\pi_{\theta})(s_t|a_t) A(s_t, a_t) \quad (20)$$

Or

$$A(s_t, a_t) = r(s_t, a_t, s_{t+1}) = \nu V(s_{t+1}) - V(s_t) \quad (21)$$

3.10 Deep Deterministic Policy Gradient (DDPG)

DDPG [24] is used to push toward profit maximization weave. DDPG combines Q-learning approaches and gradient strategies and uses neural networks as approximation functions. Unlike Reinforcement Learning (DQN), which learns using Q-values of tables, and suffers from exponential deceleration with increasing feature space [9], DDPG on-learns directly from observations through a gradient of strategies.

At each step, the DDPG agent performs the action a_t in the state s_t receives reward r_t and goes to state s_{t+1} . Transitions (s_t, a_t, s_{t+1}, r_t) are stored in buffer R . A blocks of N transitions is restored with R , and the Q-value of y_i is updated as follows (22):

$$y_i = r_i + \nu Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'}, \theta^{Q'})), i = 1, \dots, N \quad (22)$$

The critic is updated accordingly, minimizing the loss function $L(\theta^Q)$, which is the expected difference between the target network of the critic Q' , and the current network critique of Q , that is (23):

$$L(\theta^Q) = E_{s_t, a_t, r_t, s_{t+1}} \text{buffer}[(y_i - Q(s_t, a_t|\theta^Q))^2] \quad (23)$$

DDPG effectively updates the strategy for a constant space of action, therefore suitable for trading.

3.11 Proximal Policy Optimization (PPO)

PPO is researched and used as a component in an ensemble model. PPO is used to control strategy gradient updates and guarantee that the new strategy will not differ from the previous one.

3.12 Ensemble of Models

The purpose of this work is to create an adaptive trading strategy. We use an ensemble model to choose the best method for each situation.

This work was used to combine models into an ensemble Bagging method. All three models of the ensemble train in parallel on random pieces of a sample of one training dataset. Choice of ensemble result-averaging. The image (see Fig. 4) schematically shows how this method works.

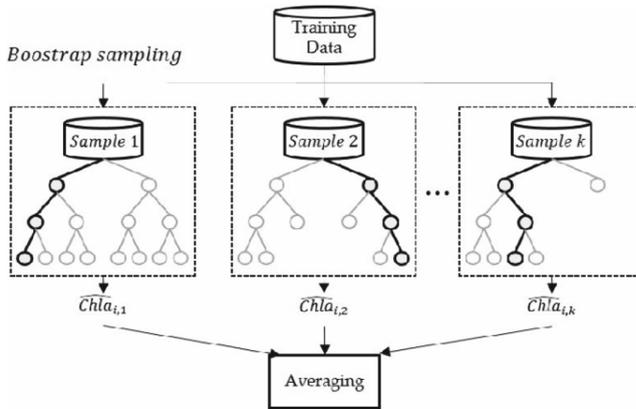


Fig. 4. Bagging

4 Experiment and Results

4.1 Training of Individual Models of the Ensemble

Using ready-made, trained models from the OpenAI Gym environment does not provide manual tuning of hyperparameters for the used models (PPO, A2C, DDPG), so most of the experiments in this work rely on direct testing of ready-made models and ensemble models, and comparable with selected standards (DOW index fund, minimization strategy variability) (see Fig. 5).

Figure 5 shows the comparative characteristics of the component ensemble models according to the metrics of the Sharpe coefficient.

Table 1 made an additional column, the best model for another period by the Sharpe coefficient.

4.2 Training and Analysis of the Ensemble

When analyzing an ensemble, it is necessary to check whether the costume is better than the used component models because otherwise, the benefits of combining models are missing (see Fig. 6).

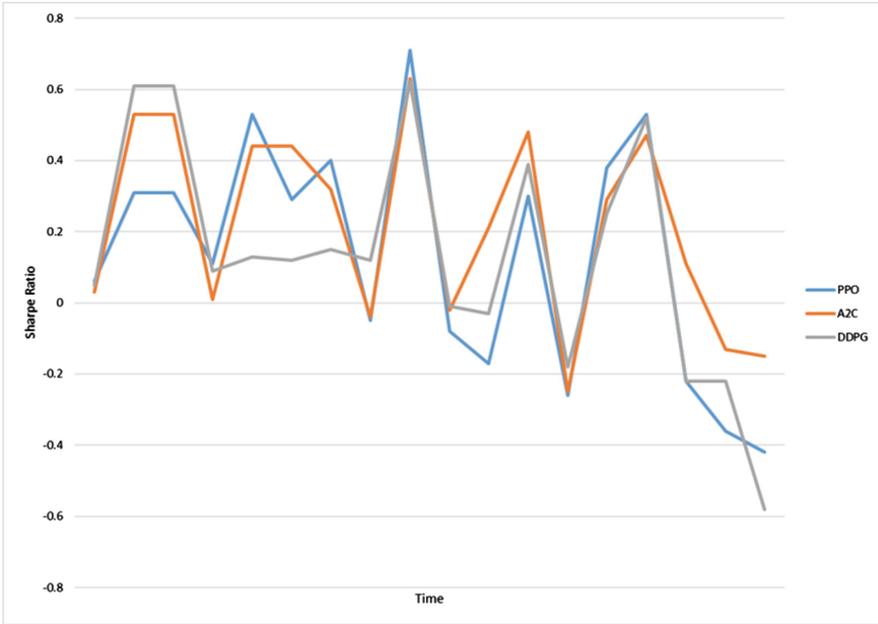


Fig. 5. Sharpe coefficient for ensemble models

Table 1. Sharpe coefficient for ensemble models.

Interval	PPO	A2C	DDPG	The bet model
2016/01–2016/03	0.06	0.03	0.05	PPO
2016/04–2016/06	0.31	0.53	0.61	DDPG
2016/10–2016/12	0.11	0.01	0.09	PPO
2017/01–2017/03	0.53	0.44	0.13	PPO
2017/04–2017/06	0.29	0.44	0.12	A2C
2017/07–2017/09	0.4	0.32	0.15	PPO
2017/10–2017/12	-0.05	-0.04	0.12	DDPG
2018/01–2018/03	0.71	0.63	0.62	PPO
2018/04–2018/06	-0.08	-0.02	-0.01	DDPG
2018/07–2018/09	-0.17	0.21	-0.03	A2C
2018/10–2018/12	0.3	0.48	0.39	A2C
2019/01–2019/03	-0.26	-0.25	-0.18	DDPG
2019/04–2019/06	0.38	0.29	0.25	PPO
2019/07–2019/09	0.53	0.47	0.52	PPO
2019/10–2019/12	-0.22	0.11	-0.22	A2C
2020/01–2020/03	-0.36	-0.13	-0.22	A2C
2020/04–2020/05	-0.42	-0.15	-0.58	A2C

```
run_ensemble_strategy(df, unique_trade_date, rebalance_window, validation_window)

====Trading from: 20200406 to 20200707
previous_total_asset:1642563.8331977944
end_total_asset:1651496.8197026039
total_reward:8932.986504809465
total_cost: 707.3734225074006
total trades: 93
Sharpe: 0.2836714121593213
Ensemble Strategy took: 150.1863147139549 minutes
```

Fig. 6. Example of terminal output during training



Fig. 7. Cumulative income for various investment strategies, namely PPO, DDPG, A2C, ensemble model, minimal variability, and Dow Jones Industrial Average (DJIA), with an initial investment of \$1,000,000.

Table 2. Portfolio indicators for various investment strategies (PPO, DDPG, A2C), ensemble model, minimal variability and Dow Jones Industrial Average (DJIA)

2016/01/04–2020/05/08	Ensemble model	PPO	A2C	DDPG
Cumulative return	70.4%	83.0%	60.0%	54.8%
Annual return	13.0%	15.0%	11.4%	10.5%
Annual volatility	9.7%	13.6%	10.4%	12.3%
Sharpe ratio	1.3	1.1	1.12	0.87
Max drawdown	-9.7%	-23.7%	-10.2%	-14.8%

Figure 6 shows an example of terminal output, and the fragment used a program that trains an ensemble. When analyzing a costume, it is necessary to check whether the chorus is better than the used component models because otherwise, the benefits of combining models are missing.

Model for a specific period (see Fig. 7). All removed data are stored for further analysis.

Figure 7 shows the cumulative return for different investment strategies compared to the ensemble model and the selected standards.

4.3 Behavior of the Ensemble Model with High Market Turbulence

As part of testing the generalization of model trading skills, an ensemble was chosen to simulate a situation of high turbulence. The agent must learn to minimize costs and choose less risky actions in this situation. This situation was selected as a test sample period from January to May 2020.

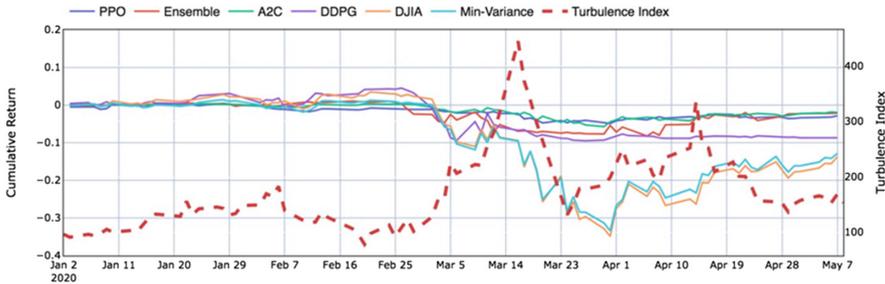


Fig. 8. Cumulative income for various investment strategies, namely PPO, DDPG, A2C, ensemble model, minimal variability, and DJIA, during the period of high turbulence index (Beginning of 2020).

Figure 8 shows the cumulative income for different investment strategies with solid multicolored lines. Starting from March, you can see a sharp decline in revenue for Minimum Variability strategies and DJIA (DOW index fund). At this time, the market reacted to COVID-19, and a sharp increase in turbulence was noticeable. The ensemble model and its components show the best results, losing the least money in this and subsequent periods.

5 Discussion

In Fig. 5, it can be seen that some models behaved differently at different sampling intervals. This may mean that each model is adapted under specific market situations. The optimal combination of models will improve trading results due to using the best model for the position.

In Fig. 6, it is seen that the ensemble model receives many times better results than DJIA and a portfolio of minimal variability. From Table 2 as well it can be seen that the ensemble achieved a higher Sharpe ratio (1.30) than DJIA (0.47) and minimal variability (0.45). This means annual income is higher, although the investment risks are lower, demonstrating that the ensemble model optimizes the balance of risk and return better than the models.

Figure 7 shows that the ensemble model and the three-component models are well applied to market collapse conditions, which means that models have learned to use the turbulence index for early stock sales and minimization losses during, for example, the collapse of the stock market in March 2020. We can change the threshold of the turbulence index to regulate risk aversion.

The results show that the ensemble model is better than the three separate algorithms. The Dow Jones average and the portfolio minimize fluctuations within the Charpy ratio by balancing risk and return.

- Ensemble by the method of Bagging deep neural networks demonstrates better results than modern analysis methods of the securities market.
- The developed environment for simulating the securities market is an adequate model for training agents.
- Begging as an ensemble method effectively combines actor-critic models, improving results.
- Ensemble and state-of-the-art models are better adapted to market conditions than 70% of people and are the two most popular strategies.

6 Conclusions

This paper explores the potential of using acting-critical models as an ensemble. Models such as Proximal Policy Optimizer (PPO), Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG) were used to teach trading strategy. For the model to adapt to different situations, Begging from three models in the ensemble was used.

However, it would be interesting to study the empirical estimates of our algorithms in such conditions to evaluate their applicability in such scenarios. There are several other ways in which our results are limited and offered for future work. First, it is a question of the speed of convergence. Ideally, we would like to get analytical results. Still, it would be helpful to conduct a thorough empirical study, changing the parameters and graphs in a more advanced and complex way than we have done here. Second, the algorithms could be extended to include acceptability traces and least-squares methods. An analysis should be performed to study the sensitivity of the submitted algorithms to various system parameters and settings. As discussed earlier, the former seems simple, while the latter needs more virtual extensions. Next, research is needed on the choice of essential functions for the critic to estimate the policy gradient reasonably. Finally, applying these ideas and algorithms to a real problem is necessary to assess their ultimate usefulness.

References

1. Agarwal, A., Kakade, S.M., Lee, J.D., Mahajan, G.: On the theory of policy gradient methods: optimality, approximation, and distribution shift (2019). <https://doi.org/10.48550/ARXIV.1908.00261>, <https://arxiv.org/abs/1908.00261>
2. Agarwal, A., Kakade, S.M., Lee, J.D., Mahajan, G.: Optimality and approximation with policy gradient methods in Markov decision processes. In: Proceedings of Thirty Third Conference on Learning Theory, pp. 64–66. PMLR (2020), <https://proceedings.mlr.press/v125/agarwal20a.html>

3. Alberg, D., Shalit, H., Yosef, R.: Estimating stock market volatility using asymmetric GARCH models. *Appl. Finan. Econ.* **18**(15), 1201–1208 (2008). <https://doi.org/10.1080/09603100701604225>, <http://www.tandfonline.com/doi/full/10.1080/09603100701604225>
4. Assran, M., Romoff, J., Ballas, N., Pineau, J., Rabbat, M.: Gossip-based actor-learner architectures for deep reinforcement learning (2019). <https://doi.org/10.48550/ARXIV.1906.04585>, <https://arxiv.org/abs/1906.04585>
5. Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., Kautz, J.: Reinforcement learning through asynchronous advantage actor-critic on a GPU (2016). <https://doi.org/10.48550/ARXIV.1611.06256>, <https://arxiv.org/abs/1611.06256>
6. Bhatnagar, S., Sutton, R.S., Ghavamzadeh, M., Lee, M.: Natural actor-critic algorithms. *Automatica* **45**(11), 2471–2482 (2009). <https://doi.org/10.1016/j.automatica.2009.07.008>, <https://linkinghub.elsevier.com/retrieve/pii/S0005109809003549>
7. Boyko, N.: Application of mathematical models for improvement of “cloud” data processes organization. *Math. Model. Comput. Sci. J. Comput. Prob. Electrotech.* **3**(2), 111–119 (2016). <https://doi.org/10.23939/mmc2016.02.111>
8. Boyko, N., Kmetyk-Podubinska, K., Andrusiak, I.: Application of ensemble methods of strengthening in search of legal information. In: Babichev, S., Lytvynenko, V. (eds.) *ISDMCI 2021. LNDECT*, vol. 77, pp. 188–200. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-82014-5_13
9. Busoniu, L., Bruin, T., Tolic, D., Kober, J., Palunko, I.: Reinforcement learning for control: performance, stability, and deep approximators. *Ann. Rev. Control* (2018). <https://doi.org/10.1016/j.arcontrol.2018.09.005>
10. Chague, F., De-Losso, R., Giovannetti, B.: Day trading for a living? *SSRN Electron. J.* (2019). <https://doi.org/10.2139/ssrn.3423101>
11. Chong, T., Ng, W.K., Liew, V.: Revisiting the performance of MACD and RSI oscillators. *J. Risk Finan. Manag.* **7**(1), 1–12 (2014). <https://doi.org/10.3390/jrfm7010001>, <http://www.mdpi.com/1911-8074/7/1/1>
12. Čermák, M., Malec, K., Maitah, M.: Price volatility modelling - wheat: GARCH model application. *Agris On-line Pap. Econ. Inf.* **09**(04), 15–24 (2017). <https://doi.org/10.7160/aol.2017.090402>
13. Gurrib, I.: Performance of the average directional index as a market timing tool for the most actively traded USD based currency pairs. *Banks Bank Syst.* **13**(3), 58–70 (2018). [https://doi.org/10.21511/bbs.13\(3\).2018.06](https://doi.org/10.21511/bbs.13(3).2018.06)
14. Kumar, H., Koppel, A., Ribeiro, A.: On the sample complexity of actor-critic method for reinforcement learning with function approximation (2019). <https://doi.org/10.48550/ARXIV.1910.08412>, <https://arxiv.org/abs/1910.08412>
15. Li, H., Dagli, C.H., Enke, D.: Short-term stock market timing prediction under reinforcement learning schemes. In: *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 233–240 (2007). <https://doi.org/10.1109/ADPRL.2007.368193>
16. Moody, J., Saffell, M.: Learning to trade via direct reinforcement. *IEEE Trans. Neural Netw.* **12**, 875–889 (2001). <https://doi.org/10.1109/72.935097>
17. Neuneier, R.: Optimal asset allocation using adaptive dynamic programming. In: *Conference on Neural Information Processing Systems*, vol. 8, pp. 952–958. MIT Press (1996). <https://doi.org/10.5555/2998828.2998962>
18. Qiu, S., Yang, Z., Ye, J., Wang, Z.: On finite-time convergence of actor-critic algorithm. *IEEE J. Sel. Areas Inf. Theory* **2**(2), 652–664 (2021). <https://doi.org/10.1109/JSAIT.2021.3078754>, <https://ieeexplore.ieee.org/document/9435807/>

19. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017). <https://doi.org/10.48550/ARXIV.1707.06347>, <https://arxiv.org/abs/1707.06347>
20. Shen, J., Shafiq, M.O.: Short-term stock market price trend prediction using a comprehensive deep learning system. *J. Big Data* **7**(1), 66 (2020). <https://doi.org/10.1186/s40537-020-00333-6>, <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6>
21. Sun, S., Wang, R., An, B.: Reinforcement learning for quantitative trading. *arXiv:2109.13851* [cs, q-fin] (2021), <http://arxiv.org/abs/2109.13851>, *arXiv: 2109.13851*
22. Thomas, G.F.: Reinforcement learning in financial markets - a survey. Technical Report 12/2018, Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics (2018). <https://www.econstor.eu/bitstream/10419/183139/1/1032172355.pdf>
23. Xu, T., Wang, Z., Liang, Y.: Improving sample complexity bounds for (natural) actor-critic algorithms (2020). <https://doi.org/10.48550/ARXIV.2004.12956>, <https://arxiv.org/abs/2004.12956>
24. Xu, T., Wang, Z., Zhou, Y., Liang, Y.: Reanalysis of variance reduced temporal difference learning (2020). <https://doi.org/10.48550/ARXIV.2001.01898>, <https://arxiv.org/abs/2001.01898>
25. Yang, Z., Zhang, K., Hong, M., Basar, T.: A finite sample analysis of the actor-critic algorithm. In: 2018 IEEE Conference on Decision and Control (CDC), pp. 2759–2764. IEEE, Miami Beach (2018). <https://doi.org/10.1109/CDC.2018.8619440>, <https://ieeexplore.ieee.org/document/8619440/>
26. Zhang, Y., Yang, X.: Online portfolio selection strategy based on combining experts' advice. *Comput. Econ.* **50**(1), 141–159 (2017). <https://doi.org/10.1007/s10614-016-9585-0>, <https://doi.org/10.1007/s10614-016-9585-0>



Mathematical and Computer Model of the Tree Crown Ignition Process from a Mobile Grassroots Fire

Oksana Karabyn , Olga Smotr^(✉) , Andrij Kuzyk , Igor Malets ,
and Vasy Karabyn 

Lviv State University of Life Safety, Lviv, Ukraine
olgasmotr@gmail.com, {a.kuzyk,v.karabyn}@ldubgd.edu.ua

Abstract. Forest fires cause great damage to the environment and are difficult to predict and eliminate. Mathematical models of temperature regimes of forest fires can create a scientific basis for improving the effectiveness of firefighting measures. The study of fire propagation processes and temperature regimes during a fire is an urgent task that many researchers are engaged in. The processes of heat exchange in the process of fire are very complex and include many factors. For this reason, they are difficult to model. The equations of heat transfer processes themselves are also difficult to solve and obtain analytical solutions. The aim of this work is to create a mathematical model of the ignition of a needle from the flame of a moving ground fire. The solution of the set tasks was carried out by the method of direct integration. The model of the process of heating the lower part of the tree crown due to the moving grass fire described in this paper has the advantage that it is based on mathematical calculations that can be easily performed using the mathematical software package Maple. The program in the Maple package is offered, which, depending on the input data: the size of the fire area, the speed of fire spread, the distance from the ground to the bottom of the crown, calculates the air temperature. These results can be used to calculate and assess the operational situation during forest fires.

Keywords: Convective heat transfer · Flash point · Flame of a moving ground fire · Forest fire · Modeling of forest fires

1 Introduction

Forest fires, which occur in most countries of the world, cause irreparable damage to the environment and the economy. Every year, devastating forest fires break out in Spain, Portugal, Australia, Greece, the United States, and other countries. Due to the sharp warming of the planet and climate change, the scale of forest fires, the speed of their spread, and the damage they cause is catastrophic. For example, in July 2018 - a fire in Greece, killed about a hundred people and more

than 180 victims [1]; in September 2020 - the area of forest fire in California (USA) was more than 800 thousand hectares, more than 3300 buildings were destroyed, 14000 firefighters were involved in the elimination of this fire [2]; in August 2021 - forest fires had flared up and become catastrophic, uncontrollable in Greece, Turkey, and other Mediterranean countries [3]; in January 2022 - a devastating catastrophic forest fire in Colorado (USA) destroyed several hundred buildings in just a couple of hours. Unfortunately, Ukraine is no exception [5]. Only since 2022, about 90 forest fires with an area of more than 400 hectares have been eliminated in Ukraine [4].

Under such conditions, the study of the processes of forest fires becomes very relevant. To effectively organize the processes of forest fire prevention, by anticipating possible directions of its spread and effective organization of forest fire fighting, it is necessary to understand the course of forest fire and identify natural and anthropogenic factors that have the most significant impact on its development and spreading. The experimental study of forest fire on a physical model is expensive and almost unrealistic. The most appropriate is the mathematical modeling of the development process and spread of forest fires, to study them. Which in turn cannot be done without using the capabilities of modern information technology.

2 Literature Review

In scientific periodicals, we find thousands of publications related to this issue. Each of these publications contributes to the prevention of this scourge. Many works are devoted to computer modeling of fire contours and fire propagation [6, 7, 10, 11, 16, 17]. In particular, in [6] the author considers each point of the forest fire contour in space separately, regardless of neighboring points, however, such a model for constructing the movement of the forest fire contour can be reliable only when describing convex contours. In [17] the author uses the method of predicting the shape of the forest fire contour, which is based on the construction of an extrapolated image of a certain stage of forest fire development using extrapolation formulas and available images of previous stages of its development. However, over time, the shape of the forest fire circuit completely “forgets” the previous form of ignition. In [7] the author investigates the relationship between the rate of propagation of the forest fire contour and the effects of flame geometry and slope. The relationships between flame geometry and Froude’s convection number have also been studied in [15, 18, 19]. In particular, experimental testing of these models was carried out in [21–23].

The most well-known simulation models of forest fires are Rothermel [8] and Wang Zhengfei [13, 25]. However, for proper operation, they require a large number of input parameters, which, as a rule, in the conditions of rapid large-scale forest fires are absent. In addition, the Zhengfei simulation model [25] is valid only when modeling the spread of forest fire in areas where the terrain has no slope of more than 60° . Particular attention should be paid to works on mathematical modeling of heat transfer and heat and mass transfer processes during

forest fires [9, 12, 20, 24]. In particular, in [12] a mathematical model of the process of heating the needle due to radiant heat radiation from the surface in the form of a rectangle can set the time of its heating to the autoignition temperature. In [9], a mathematical model for predicting the spread of the landscape forest fire contour was developed, taking into account three types of heat exchange: heat conduction, convective heat exchange, and heat radiation. It should be noted that a number of the above models meet the requirements for validation of forest fire spreading models specified in [14].

Most of the above models require a large number of input parameters that are not always available and perform complex mathematical calculations. Obviously, without using modern integrated software systems and software packages to automate mathematical calculations they are unusable.

In addition, in the conditions of ephemerality and uncertainty of the development of large-scale forest fire, the head of forest fire extinguishing, to begin the process of forest fire extinguishing it is necessary to obtain at least its general characteristics as soon as possible. Therefore, in order to optimize the time of obtaining a model of forest fire, we neglect a number of input parameters that can be specified in the process of forest fire and build a simplified mathematical model of burning needles on tree canopies from the flames of mobile grassland fire using Maple.

3 Problem Statement

Many scientific publications are devoted to the problems of modeling the spread of fire. It is necessary to single out works devoted to modeling of fire torches, construction of simulation models of fires. Calculation of temperature regimes during an open fire is associated with great technical difficulties. Such models need to take into account a large number of parameters and conditions and three types of heat transfer: thermal conductivity, convection, radiation. As a result, the mathematical model consists of differential equations in partial derivatives of higher orders. For such models it is almost impossible to obtain solutions in analytical form. In addition, such models are very difficult to algorithmize and program. We propose to consider a model based on the laws of radiation without taking into account convective heat transfer. This model is much simpler because it is described by a first-order differential equation.

The aim of our study is to create a mathematical model of heat transfer during a ground fire, which can be used to set the air temperature at the desired height depending on the time, speed of fire and the size of the fire area. To create this model, we rely on the known equations of the laws of heat transfer in the process of heating the surface of a certain area to a certain temperature. In our model, we want to take into account that the area of the fire is moving at a constant speed. For convenience of calculations, we believe that the fire area has a rectangular shape. In our model, we introduce a coordinate system that allows you to take into account the dynamics of the fire at a constant speed. In the process of modeling, we expect the opportunity to obtain the dependence of

temperature on time. The differential equation includes the angular irradiation coefficient, which is also a function of time. In this situation, there are difficulties with integration. As a result, we obtain the solution of the differential equation in implicit form. Modern software allows you to perform calculations and get point results in the required number with a high speed of calculations. We have created an algorithm, which is later presented by pseudocode in our article. We performed calculations using the Maple mathematical package. In the first stage, integration steels are calculated for each value of the distance from the fire to the lower part of the crown, in the second stage, irradiation coefficients and temperature values are calculated.

4 Matherial and Methods

Consider the theoretical foundations of our model. To heat some surface with area S with temperature T , irradiated from the surface of the rectangle of constant temperature T_1 (consider the surface flat) during time dt , the amount of heat will be spent

$$Q = \sigma\varepsilon(T_1^4 - T^4) \tag{1}$$

From another point of view, heating this surface by a magnitude dT determined by the formula

$$Q = cmdT \tag{2}$$

where m is the weight of surface, unit of measurement - kilogram σ is the Stefan-Boltzmann constant, ε is the emissivity, ψ is the angular irradiation coefficient (by Lambert's law), c is the specific heat, $\frac{Joule}{kg \cdot K}$. We assume that all the heat received by the radiation will be spent on heating the surface. Equate the right parts of formulas (1) and (2). We obtain a differential equation:

$$cmdT = \sigma\varepsilon(T_1^4 - T^4)\psi(t)Sdt. \tag{3}$$

This equation describes the process of heating the surface to temperature $T = T(t)$ over time t .

The initial condition for this equation is the condition:

$$T(0) = T_0$$

The radiation coefficient ψ for an element of an area dS and a flat rectangle parallel to it , if the normal to the element passes through the angle of this rectangle is determined by the formula (Fig. 1)

$$\psi = \frac{1}{2\pi} \left(\frac{a}{\sqrt{a^2 + r^2}} \arctan \frac{b}{\sqrt{a^2 + r^2}} + \frac{b}{\sqrt{b^2 + r^2}} \arctan \frac{a}{\sqrt{b^2 + r^2}} \right) \tag{4}$$

For the case if the irradiated area dS is located opposite the center of the rectangle, the irradiation coefficient is determined by the formula

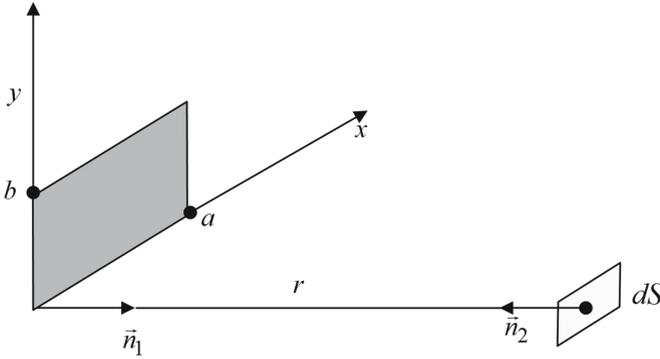


Fig. 1. Irradiation of a stationary surface

$$\psi = \frac{2}{\pi} \left(\frac{a}{\sqrt{a^2 + 4r^2}} \arctan \frac{b}{\sqrt{a^2 + 4r^2}} + \frac{b}{\sqrt{b^2 + 4r^2}} \arctan \frac{a}{\sqrt{b^2 + 4r^2}} \right) \quad (5)$$

For the case if the irradiated area dS is located opposite the point of the rectangle, which has coordinates $(x; y)$, the irradiation coefficient is determined by the formula

$$\begin{aligned} \psi = & \frac{2}{\pi} \left(\frac{y}{\sqrt{y^2 + 4r^2}} \arctan \frac{x}{\sqrt{y^2 + r^2}} + \frac{x}{\sqrt{x^2 + y^2}} \arctan \frac{y}{\sqrt{x^2 + r^2}} \right) \\ & + \frac{1}{2\pi} \left(\frac{y}{\sqrt{y^2 + r^2}} \arctan \frac{b-x}{\sqrt{y^2 + r^2}} + \frac{b-x}{\sqrt{(b-x)^2 + y^2}} \arctan \frac{y}{\sqrt{(b-x)^2 + r^2}} \right) \\ & + \frac{1}{2\pi} \left(\frac{a-y}{\sqrt{(a-y)^2 + r^2}} \arctan \frac{x}{\sqrt{(a-y)^2 + r^2}} + \frac{x}{\sqrt{x^2 + r^2}} \arctan \frac{y}{\sqrt{x^2 + r^2}} \right) \\ & + \frac{1}{2\pi} \left(\frac{a-y}{\sqrt{(a-y)^2 + r^2}} \arctan \frac{(b-x)}{\sqrt{(a-y)^2 + r^2}} \right) \\ & + \frac{1}{2\pi} \left(\frac{(b-x)}{\sqrt{(b-x)^2 + r^2}} \arctan \frac{a-y}{\sqrt{(b-x)^2 + r^2}} \right) \end{aligned} \quad (6)$$

Consider a needle or leaf, which is at a height of r , m from the ground. The surface of a flame is considered to be a horizontal rectangle a , m long and d , m wide, moving at a constant speed $x = v(t - t_0)$ in a direction perpendicular to the leaves or needles. Heat exchange between the needle or leaf and the moving surface is carried out according to formula (5). If you choose a coordinate system so that the needle is projected on a straight line passing through the center of the rectangle, parallel to the width d and impose the condition that at the time $t = 0$ the moving surface is in position $-x$ (Fig. 2), then formula (5) will look like

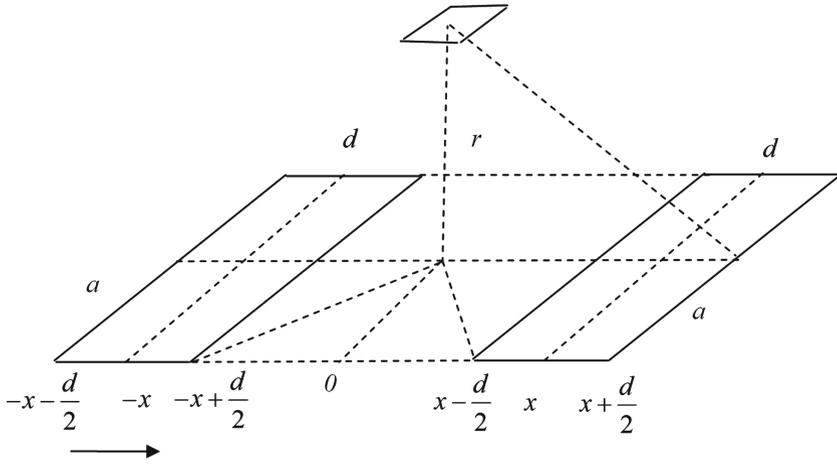


Fig. 2. Movable irradiation surface

$$\begin{aligned}
 \psi &= \frac{1}{\pi} \frac{a}{\sqrt{a^2 + 4r^2}} \arctan \frac{2x + d}{\sqrt{a^2 + 4r^2}} \\
 &+ \frac{1}{\pi} \frac{2x + d}{\sqrt{(2x + d)^2 + 4r^2}} \arctan \frac{a}{\sqrt{(2x + d)^2 + 4r^2}} \\
 &\quad - \frac{1}{\pi} \frac{a}{\sqrt{a^2 + 4r^2}} \arctan \frac{2x - d}{\sqrt{a^2 + 4r^2}} \\
 &+ \frac{1}{\pi} \frac{2x - d}{\sqrt{(2x - d)^2 + 4r^2}} \arctan \frac{a}{\sqrt{(2x - d)^2 + 4r^2}} \tag{7}
 \end{aligned}$$

Substitute in (7) $x = vt$, we obtain

$$\begin{aligned}
 \psi(t) &= \frac{1}{\pi} \frac{a}{\sqrt{a^2 + 4r^2}} \arctan \frac{2vt + d}{\sqrt{a^2 + 4r^2}} \\
 &+ \frac{1}{\pi} \frac{2vt + d}{\sqrt{(2vt + d)^2 + 4r^2}} \arctan \frac{a}{\sqrt{(2vt + d)^2 + 4r^2}} \\
 &\quad - \frac{1}{\pi} \frac{a}{\sqrt{a^2 + 4r^2}} \arctan \frac{2vt - d}{\sqrt{a^2 + 4r^2}} \\
 &+ \frac{1}{\pi} \frac{2vt - d}{\sqrt{(2vt - d)^2 + 4r^2}} \arctan \frac{a}{\sqrt{(2vt - d)^2 + 4r^2}} \tag{8}
 \end{aligned}$$

In Eq. (3), separate the variables and integrate both parts of equation. As a result, we obtain

$$\frac{cm}{S\sigma\varepsilon} \int \frac{dT}{T_1^4 - T^4} = \int \psi(t) dt \tag{9}$$

Using the method of integration by parts, we obtain the expression for the right part (9):

$$\begin{aligned}
 \Psi(t) &= \int \psi(t) dt = \\
 &= \frac{a}{2v\pi} \left(\frac{2vt+d}{\sqrt{a^2+4r^2}} \arctan \frac{2vt+d}{\sqrt{a^2+4r^2}} - \frac{1}{2} \ln \left(1 + \frac{(2vt+d)^2}{a^2+4r^2} \right) \right) \\
 &\quad + \frac{1}{2v\pi} \sqrt{(2vt+d)^2+4r^2} \arctan \frac{a}{\sqrt{(2vt+d)^2+4r^2}} \\
 &\quad + \frac{1}{2v\pi} \frac{a}{2} \ln((2vt+d)^2+4r^2+a^2) \\
 &\quad + \frac{a}{2v\pi} \frac{2vt-d}{\sqrt{a^2+4r^2}} \arctan \frac{2vt-d}{\sqrt{a^2+4r^2}} - \frac{a}{4v\pi} \ln \left(1 + \frac{(2vt-d)^2}{a^2+4r^2} \right) \\
 &\quad + \frac{1}{2v\pi} \sqrt{(2vt-d)^2+4r^2} \arctan \frac{a}{\sqrt{(2vt-d)^2+4r^2}} \\
 &\quad + \frac{a}{4v\pi} \ln((2vt-d)^2+4r^2+a^2) \tag{10}
 \end{aligned}$$

After direct integration of the left part of formula (9) we obtain

$$\int \frac{dT}{T_1^4 - T^4} = \frac{1}{4T_1^3} \left(\ln \left| \frac{T_1+T}{T_1-T} \right| + 2 \arctan \frac{T}{T_1} \right) + C \tag{11}$$

Thus we obtain the analytical expression of the general integral of the differential equation, which describes the process of heat transfer between the needles and the moving surface of the radiation.

$$\frac{cm}{S\sigma\varepsilon} \frac{1}{4T_1^3} \left(\ln \left| \frac{T_1+T}{T_1-T} \right| + 2 \arctan \frac{T}{T_1} \right) = \psi(t) + C \tag{12}$$

Equation (12) is transcendent. To find the approximate solution, we decompose the left-hand side of this equation into the Taylor series.

$$\ln \left| \frac{T_1+T}{T_1-T} \right| + 2 \arctan \frac{T}{T_1} = \frac{2T_1}{T-T_1} - \frac{2T_1^2}{(T-T_1)^2} + \frac{4T_1^3}{3(T-T_1)^3} + \frac{2T}{T_1} - \frac{T^3}{3T_1^3}$$

To find the integration constant, we substitute the initial conditions and input data. After substituting the constant in Eq. (12) we can find the value of temperature. Finally, the equation for finding the temperature has the form:

$$\frac{2T_1}{T-T_1} - \frac{2T_1^2}{(T-T_1)^2} + \frac{4T_1^3}{3(T-T_1)^3} + \frac{2T}{T_1} - \frac{T^3}{3T_1^3} = \psi(t) + C \tag{13}$$

5 Experiment, Results and Discussion

The model of temperature regimes of the lower part of the tree crown created by us has an implicit analytical solution given in formula (12). Dependence (12) is a transcendental function with respect to the variable T . We have solved this problem by decomposing the left-hand side of Eq. (12) into the Taylor series. As a result of such replacement the accuracy of the solution is lost. But the advantage is that the resulting equation is an algebraic equation of the third degree. Such equations can be solved quickly with the help of application mathematical packages that are publicly available. For this reason, the Maple software package was used to find the roots of this equation. Based on it, a program was created to calculate the integration constant and find the temperature of the needles or leaves of the tree depending on the distance from the soil surface to the bottom of the crown. The program takes into account the possibility of correcting the speed of fire, the size of the fire area, the distance from the soil surface to the bottom of the crown. The advantage of this method is the high speed of calculations, the availability of software. The input data to the problem are shown in Table 1. The flame temperature of the moving area is taken $T_1 = 900$ K. This is an average that can be replaced with another value if necessary. Assume that the ambient temperature is $T_0 = 293$ K. The program provides the ability to change the size of the fire zone and the speed at which it spreads. The speed of ground fire propagation, which is $v = 0,004$ meters per second, is taken for calculations. In addition, we have created an algorithm for calculating the temperature regime. Presented below Algorithm 1 describes the stepwise procedure of temperature calculations.

Table 1. The input data to the problem

No.	Abbr.	Assentiality	Quantity	Dimension
1	a	The width of the fire area	2	meters
2	d	The lenth of the fire area	2	meters
3	r	The height of the lower part of the tree crown	varies from 2 to 40	meters
4	v	Speed of fire spread	0,004	m/c
5	c	Specific heat	1172	$\frac{Joule}{kg \cdot K}$
6	T_0	Embient temperature	293	K
7	T_1	Flame temperature	900	K
8	m	Mass of a tree leaf	$0,062 \cdot 10^{-3}$	kg
9	ε	Emissivity	0,7	
10	σ	The Stefan - Boltzman constant	5,67	$\frac{Wt}{m^2 \cdot K^4}$
11	t	time	$[0; t_{max}]$	s

Algorithm 1: Calculation of the temperature of the lower part of the crown

Initialization:

calculate: calculate the integration constants C ;
 calculate the radiation coefficients $\psi_k(t)$;
 create an array of temperatures T ;
 set: iteration counter r -the height of the lower part of the tree crown r ;
 iteration step of counter r $step1 = 4$
 iteration counter k , iteration step of counter k $step2 = 16$
 the size of the fire area a, d ;
 speed of fire spread v ;
 specific heat c ;
 ambient temperature T_0 ;
 flame temperature T_1 ;
 mass of a tree leaf m ;
 emissivity ε ;
 the Stefan - Boltzman constant σ ;
 $a, d, v, step1, step2, T_0$ - set of input parameters, can be changed by the user, taking into account the operational situation on the fire

while $r \leq r_{max}$ **do**

calculate the integration constant C

while $k \leq t_{max}$ **do**

Calculate the angular irradiation coefficient $\psi_k(t)$;

Calculate the temperature $T_k(t)$

$k = k + step2$;

end

$r = r + step1$;

end**Return** array of temperatures T .**Table 2.** The results of the calculation of the integration constants

r	2	6	10	14	18	22
C	417,99	559,68	637,78	690,44	730,06	761,81

Table 3. Dependence of crown temperature on time for different distances from the soil surface

T	t, c										
	0	16	32	48	64	80	96	112	128	144	160
$r = 2$	909.78	909.79	909.78	909.77	909.76	909.74	909.72	909.70	909.68	909.66	909.63
$r = 6$	905.13	905.13	905.13	905.13	905.13	905.12	905.12	905.12	905.12	905.12	905.12
$r = 10$	904.91	904.91	904.91	904.91	904.91	904.91	904.91	904.91	904.91	904.91	904.90
$r = 14$	904.78	904.78	904.78	904.78	904.78	904.78	904.78	904.78	904.78	904.78	904.78
$r = 18$	904.69	904.69	904.69	904.69	904.69	904.69	904.69	904.69	904.69	904.69	904.69
$r = 22$	904.63	904.63	904.63	904.63	904.63	904.63	904.63	904.63	904.63	904.63	904.63

The results of the calculation of the integration constants for the corresponding values of the crown height are shown in Table 2. The dependence of the temperature of the lower part of the tree crown on time is also shown in Table 3. The obtained temperature indicators indicate that due to the mobile fire the lower part of the tree crown heats up to the autoignition temperature. Another important feature is the observed decrease in temperature by 4K in the transition from a height of 2 m to a height of 6 m from the soil surface. At transition from height of 6 m to height of 10 m temperature decreases only on 1 K. From a height of 10 m to a height of 22 m, the temperature remains almost stable and is 904 K.

6 Conclusions

The study of fire propagation processes and temperature regimes during a fire is an urgent task that many researchers are engaged in. The processes of heat exchange in the process of fire are very complex and include many factors. For this reason, they are difficult to model. The equations of heat transfer processes themselves are also difficult to solve and obtain analytical solutions. The model of calculation of temperature regimes of the lower part of the tree crown during a grassland fire proposed in our work is based on the first-order differential equation, the solution of which is obtained analytically. The calculation algorithm created by us consists of two nested cycles, on the basis of which the temperature array of the lower part of the crown is formed. The advantage of our model is that it is algorithmic and is available for programming the necessary calculations. The program in the Maple package is offered, which, depending on the input data: the size of the fire area, the speed of fire spread, the distance from the ground to the bottom of the crown, calculates the air temperature. According to the results of modeling with input data: the flame temperature of the moving area $T_1 = 900$ K, ambient temperature $T_0 = 293$ K, the speed of ground fire propagation $v = 0,004$ m/s, it was found that the temperature at the bottom of the crown varies from 905 K to 910 K. The model allows calculations with any input data. The advantage of our model is the possibility of primary

operational calculation of temperature regimes. The user enters the input data and the program gives the desired result. These results can be used to calculate and assess the operational situation during forest fires.

References

1. Fires in Greece: the death toll rose to 74, more than 180 injured (2018). <https://www.eurointegration.com.ua/news/2018/07/24/7084762/>
2. The fires in California covered 809 thousand hectares (2020). https://lb.ua/world/2020/09/08/465498_pozhezhi_kalifornii_ohopili_809_tisyach.html
3. Fires in southern Europe: the situation is getting worse (2021). <https://www.dw.com/uk/pozhezhi-na-pivdni-yevropy-sytuatsiia-zahostriuietsia/a-58775185>
4. The average area of forest fires in Ukraine has increased 7 times (2022). https://lb.ua/society/2022/03/29/511485_cerednya_ploshcha_lisovoi_pozhezhi.html
5. In Colorado, fires destroyed hundreds of homes within hours (2022). https://lb.ua/world/2022/01/01/502250_kolorado_pozhezhi_zruynuvali_sotni.html
6. Dorrer, G.: *Mathematical Models of Forest Fire Dynamics*. Forest Industry, Moscow (1979)
7. Egorova, V.N., Trucchia, A., Pagnini, G.: Fire-spotting generated fires. part II: the role of flame geometry and slope. *Appl. Math. Model.* **104**, 1–20 (2022). <https://doi.org/10.1016/j.apm.2021.11.010>
8. Ervilha, A., Pereira, J., Pereira, J.: On the parametric uncertainty quantification of the Rothermel's rate of spread model. *Appl. Math. Model.* **41**, 37–53 (2017). <https://doi.org/10.1016/j.apm.2016.06.026>
9. Gulida, E., Smotr, O.: Forest fire forecasting. *Fire Saf. Issues* **21**, 73–80 (2007)
10. Hramov, V., et al.: Modeling of forest fires on Grid clusters. *Comput. Tools Netw. Syst.* **21**, 47–57 (2012)
11. Kononov, M., Sudakov, O., Hramov, V.: Features of modeling of fires on clusters in Grid. *Bull. Taras Shevchenko Natl. Univ. Kyiv. Phys. Math. Sci. Collect. Sci. works* **2**, 185 (2011)
12. Kuzyk, A., Tovaryansky, V.I.: Mathematical modeling of conductive and radiative heat transfer processes during a fire in pine forests (2017)
13. Li, X., et al.: Simulating forest fire spread with cellular automation driven by a LSTM based speed model. *Fire* **5**(1), 13 (2022). <https://doi.org/10.3390/fire5010013>
14. Morvan, D.: Validation of wildfire spread models. In: Manzello, S., et al. (eds.) *Encyclopedia of Wildfires and Wildland-Urban Interface (WUI) Fires*, pp. 1031–1037. Springer, Cham (2020). https://doi.org/10.1007/978-3-319-52090-2_59
15. Nelson Jr., R.: Byram's energy criterion for wildland fires: units and equations. *Research Note INT-US Department of Agriculture* (1993)
16. Smotr, O., Grytsyuk, Y.: Modeling of forest fire contours. *Fire Secur.* **20**, 169–179 (2012)
17. Soznik, O., Kalinovsky, A.: Geometric model of the rate of spread of landscape fires and some of its consequences. In: *Applied Geometry and Engineering Graphics. Proceedings - Tavriya State Agrotechnical Academy* **4**, 94–98 (2004)
18. Sullivan, A.: Convective Froude number and Byram's energy criterion of Australian experimental grassland fires. *Proc. Combust. Inst.* **31**(2), 2557–2564 (2007)
19. Sullivan, A.L.: Inside the inferno: fundamental processes of wildland fire behaviour. *Curr. Forestry Rep.* **3**(2), 132–149 (2017). <https://doi.org/10.1007/s40725-017-0057-0>

20. Tatsiy, R.M., Pazen, O.Y., Vovk, S.Y., Ropyak, L.Y., Pryhorovska, T.O.: Numerical study on heat transfer in multilayered structures of main geometric forms made of different materials. *J. Serb. Soc. Comput. Mech.* **13**(2), 36–55 (2019). <https://doi.org/10.24874/jsscm.2019.13.02.04>
21. Weise, D.R., Biging, G.S.: Effects of wind velocity and slope on flame properties. *Can. J. For. Res.* **26**(10), 1849–1858 (1996)
22. Weise, D.R., et al.: Fire behavior in chaparral—evaluating flame models with laboratory data. *Combust. Flame* **191**, 500–512 (2018). <https://doi.org/10.1016/j.combustflame.2018.02.012>
23. Weise, D., Zhou, X., Sun, L., Mahalingam, S.: Fire spread in chaparral (2005)
24. Zelensky, K., Hazel, V.: Mathematical modeling of aerodynamics of horse forest fires. *Sci. Notes* **27**, 110–115 (2010)
25. Zhang, X., Liu, P.: Research on the improvement of Wang Zhengfei’s forest fire spreading model. *Shandong Forest. Sci. Technol.* **50**(01), 1–6 (2020)



Features of Complex Application of the Formal Method of EVENT-B for Development of Environmental Management Systems

Oleh Mashkov¹ , Oleh Ilyin² , Viktor Mashkov³ , Oleh Boskin⁴ ,
and Oksana Ohnieva⁴  

¹ State Ecological Academy of Postgraduate Education and Management,
Kyiv, Ukraine

mashkov_oleg_52@ukr.net

² State University of Telecommunications, Kyiv, Ukraine

info@dut.edu.ua

³ Jan Evangelista Purkyně University in Ústí nad Labem,
Ústí nad Labem, Czech Republic

viktor.mashkov@ujep.cz

⁴ Kherson National Technical University, Kherson, Ukraine

oksana.ognieva@meta.ua

Abstract. Scientific research is devoted to determining the areas of integrated application of the formal method Event-B for the development of environmental management systems. The practice of prevention and elimination of environmental emergencies indicates the general nature of risks in the field of environmental, industrial and occupational safety. Accordingly, the management of these risks is optimally carried out on a single basis - in the framework of the creation of environmental management systems (objects of management are considered man-made hazardous objects, ecosystem objects). The application of the method of formal verification of Model Checking and the method of specification of requirements of Event-B at synthesis of administrative ecological decisions is offered. The formalization of management decisions in the system of ecological management in the conditions of abnormal ecological situations is given.

Keywords: Formal methods · Event-B · Failure modes · Effects (and criticality) analysis · Fault tree analysis · Reliability of computer systems · Ecological management system · Ecological systems

1 Introduction

The practice of prevention and liquidation of emergencies (EM) shows the general nature of risks in the field of environmental, industrial and occupational safety. Accordingly, the management of these risks is optimally carried out on a

single basis - within the framework of integrated management systems (IMS), in particular - HSE-management (management of labor protection, industrial and environmental safety). ISM has become a “new word” in approaches to managing organizations in various industries and levels. The creation of such management systems involves the unification of approaches to ensuring activities in different directions. The organization and support of ISM promotes equality of systems and decision-making contribute to the continuous improvement of the organization. It is a powerful tool to increase efficiency, obtain a synergistic effect and save all kinds of resources [1–4]. It is known that the formal method of software development (B-Method) was developed by Jean-Raymond Abrial in the UK and France. This method has been used in Europe to create systems that are critical to reliability (a mistake that poses a threat to people and other resources). B-Method is considered as a set of mathematical technologies for the specification, design and implementation of software components. Another method called Event-B has recently been proposed. It is seen as an improvement on the B-Method (also known as the classic B). Event-B has a simpler syntax that makes it easier to learn and use. The tools that support it are defined as the Rodin platform [6, 9].

At present, the possibility of integrated application of the formal Event-B method for the development of environmental management systems is not scientifically substantiated.

The rest of the paper is structured in a following way. The Sect. 2 reviews the literature on the integrated application of the formal Event-B method for the creation of environmental management systems. The Sect. 3 presents the results of a study of the application of the formal method Event-B in environmental management systems. In Sect. 4, computer simulations are performed and the effectiveness of formal methods is evaluated. The Section “Conclusions” summarizes the research results and indicates the direction for the future work.

2 Review of Literature

For the development of environmental management systems, it is proposed to use formal methods [5]. The use of formal methods involves the use of so-called invariants, which are designed to monitor compliance with the properties associated primarily with security functions.

The invariant allows you to check the implementation of one or more properties of the environmental management system and the corresponding requirements. It should be noted that all the requirements for the developed system, both functional and non-functional, are potentially invariant. However, not all requirements can be easily presented in a form suitable for use as invariants in formal methods. In addition, not all requirements should be presented in the form of invariants. The very task of generating (searching, synthesizing) invariants is not formal. Its decision depends on the experience and qualifications of the developer, expert and is in some sense the nature of art, which to some extent contradicts the very essence of formal methods. At the same time, if

there is sufficient information to calculate the metrics of individual invariants, the assessment of the quality of the system of invariants can be a fairly simple and formal procedure [11, 13, 20].

Synthesis of invariants can be carried out in two ways: based on the analysis of initial requirements (specifications) for the environmental management system - in its development and step-by-step details, starting from verbal specification and ending with the generation of program code with proof of project correctness; based on the analysis of the project of the environmental management system and identification of its properties that can be represented by invariants in the verification of the completed project.

The first method can be implemented in the following General sequence:

- many requirements for the environmental management system are normalized (priorities are selected and specified);
- for each of the requirements and/or their subset is determined by the covering invariant or covering the subset of invariants;
- the problem of covering the requirements of many invariants is solved and the minimum systems of invariants are determined;
- the optimal system of invariants according to the set criteria is chosen.

The second method is implemented either in the same general sequence as the first method, or its modification, based on the formation of many invariants not according to the requirements of the specification, but according to the results of project analysis and detection of constant relationships. Such relations (invariants) may include, for example, semantic invariants, which control the correct-ness of the dimensionality of variables.

At present, the development of ecological management systems and their achievement of such a level of perfection that would ensure the harmonious development of production and natural resources, would reconcile the “goals” of human management with the “goals” of self-regulation of natural ecosystems and avoid environmental degradation. The ability of the management system to quickly restructure to achieve new goals is characterized by its dynamism, maneuverability, and hence efficiency in meeting the needs and security of the state, responding to changing geo-ecological situation. The new paradigm of sustainable development has set new goals both in the field of public administration and in the field of environmental management. Thus, there is a contradiction between the new targets and the old structure, not designed to perform them. Therefore, as a result of the system analysis, the directions of complex application of the formal Event-B method for the development of environmental management systems should be formulated.

At present, formal methods for creating and analyzing environmental management systems are not widespread. This is primarily due to the lack of effective technology for analyzing the behavior of man-made ecological systems in emergencies, emergencies and catastrophic situations [7, 8, 10, 12, 14–19].

For critical application systems, it is also necessary to determine the means of failure-safety, i.e. means of accident prevention, even in the event of any failure, despite the means of failure (to bring the system to a protected state of failure).

The joint use of the formal method of developing Event-B systems and analyzing their reliability (FMECA, FTA) allows to expand the practical application of Event-B, expanding its capabilities to systems critical to failure due to both structural defects and physical defects.

The article presents the results of sharing the formal Event-B method, the method of analysis of types and consequences of critical failures FME (C) A, as well as the method of analysis of FTA failure trees on the example of the environment. System of mental management in emergencies, emergencies and catastrophic situations. It should be recognized that to date there are no objective reasons, both technical and technological, which would prevent the widespread practical use of formal methods. The use of formal methods requires the involvement of more qualified developers with serious mathematical training and practical skills to express the requirements for the system and the logic of its work with the help of mathematical apparatus. The least formalized processes of formal methods are the refinement of the system model, the formulation of invariants and the formation of the system of invariants. The use of methods similar to Event-B threatens the transition from the problem of “infinite testing” to the problem of “infinite detail” of the model being developed. Other problems of this class of formal methods are: the process of detailing the informal model, limited support for complex data structures and operations with numbers that are not many natural numbers, lack of proof of the completeness of the invariant system used, the need for choice, complexity. Construction and verification of evidence of means of automatic execution. The complexity of building a model of the analyzed system in some cases may be comparable in complexity to the development of the system itself. Significant simplification of the model leads to a decrease in its adequacy. It should also be noted that the joint use of Event-B and Model Checking methods in one project using one invariant system is excessive and impractical.

Methodological and General Scientific Significance. When designing and analyzing complex systems, it is often necessary to describe models of these systems. One approach to this task is to use the formal Event-B method [1, 3, 7, 9, 10, 12, 20].

Critical security systems, errors in which can lead to death or injury, large financial losses and damage to the environment, make increased demands on their correctness. One way to increase confidence in the correctness is to model the system and its requirements and bring them into line with the use of formal methods. The proposed approach to the use of the Event-B method is to formalize and prove the correctness of the model of management of access rights and information flows of the special-purpose operating system Astra Linux Special Edition. The model was fully formalized and verified, which allowed to identify and correct a number of inaccuracies in its original textual description. As a result, we can say that the use of formal methods has the following advantages: finding errors that would not otherwise be found; increase confidence in the correct operation of the system; solving the problem of system support with further changes and expansions in its description.

3 Material and Methods

It is known that the Event-B method is based on the use of abstract machine notation AMN (Abstract Machine Notation) [2,4,6,14] and allows to obtain program code by developing and gradually detailing the formal specification of discrete systems. Event-B formalizes the process of describing the properties and dynamic behavior of systems, as well as provides control over the observance of these properties in the process of functioning on the basis of the prerequisite mechanism.

The use of Event-B can improve the quality of system requirements (reduce the likelihood of defects in requirements), as well as significantly reduce or eliminate design defects, ensuring the correct behavior of the system within the system used invariants. The key principle of the formal Event-B method is refinement or detailing, which involves a gradual transition from a more abstract system model to a more specific one.

The general approach to the development of formal specifications using Event-B notation is the gradual detailing of the system while preserving its basic properties, which are recorded using invariants. As part of the formal Event-B method, the requirements for safety, acceptable degradation and temporal characteristics are usually converted into invariants, i.e. conditions that must be constantly met. Other requirements are reflected in the development of the system model. The design phase is iterative. Each iteration details the system model and proves its correctness. At the same time, there is no clear boundary between the requirements specification stage and the design stage, and the need for autonomous and partially integrated testing is eliminated. The transition to the coding stage is ideally carried out by automatically generating (using appropriate utilities) program code from the last detailed system model.

The purpose of the Event-B application is to minimize or completely eliminate design defects within the invariant system used. At the same time, the method is poorly suited for creating systems that are resistant to physical failures of the elements [1,3,9,20].

3.1 Features of Application of the Method of Formal Verification Model Checking

When using the Model Checking method, instead of testing the system, a formal verification of its model is performed. In addition, two additional processes are added [15]:

1. modeling, which is to bring the designed system to a formal form that would be acceptable for tools for verifying program models, such as SPIN or SMV [10]. When developing complex software packages, it is almost impossible to obtain the Krypka model with a finite number of states directly from the program code. In this case, an abstraction of the system is required;
2. formalization of requirements, which presents the initial requirements for the system in the form of expressions of temporal logic, for example, using the

linear time logic LTL (Linear Temporal Logic) or logic tree CTL (Computational Tree Logic).

In this context, the joint application of formal methods of system development and formal methods of analysis of their functional stability is relevant. Of particular interest is the joint application of the Event-B Requirements Specification Method, the FME (C) A Critical Failure Type and Consequence Analysis Method and the FTA Failure Tree. This conclusion is logical given the notion of completeness and minimality of sets of invariants, since FME (C) A-tables (FTA-trees) are systematized information about possible failures. The aim of this article is to study and practically illustrate the possibility of sharing the formal Event-B method, as well as tabular graph methods for analyzing the functional stability of FME (C) A and FTA on the classic example of environmental management system [1, 16, 19].

3.2 The Essence and Basic Principles of the Event-B Requirements Specification Method

It is known that the Event-B method is based on the use of abstract machine notation AMN (Abstract Machine Notation) for formal software development. This method formalizes the process of describing the properties and dynamic behavior of systems, provides control over the observance of these properties in the process of functioning on the basis of the prerequisite mechanism, and allows to obtain program code by developing and phasing detailed formal system specifications. When using the formal method Event-B, the system specification is presented in the form of its formal model, the main elements of which are:

- a set of system variables (variables), the specific value of which reflects a certain state of the system (state);
- context, presented as a set of system constants;
- invariants - a set of properties (conditions), the truth of which must always be observed in the process of system operation;
- events that occur inside or outside the system and that move the system from one state to another by the system performing certain operations that change the values of system variables in response to each event;
- a set of preconditions (guards) for each event that prohibit its occurrence, if as a result of the system's response to this event there is a violation of invariants.

The key features of Event-B are step-wise refinement of the system model and automatic proof of its correctness. Detailing involves a gradual transition from a more abstract model of the system to a more specific one based on adding new events or changing existing ones, adding new operations, variables, pre- and post-conditions, invariants and/or constants. Proof of correctness is performed by listing all events and invariants, and automatically-performed mathematical proof that when each event, taking into account the mechanism of preconditions, there is no violation of any invariant.

3.3 The Essence and Basic Principles of Constructing FTA Failure Trees

Practice shows that the occurrence and development of environmental and man-made accidents is usually characterized by a combination of random local events that occur with different frequency at different stages of the accident (equipment failures, human errors, external influences, destruction, intoxication, etc.). Logical and graphical methods of fault tree analysis (FTA) are used to identify causal relationships between these events. The FTA failure tree construction method can be used to analyze the possible causes of environmental accidents and catastrophes and to calculate the frequency of their occurrence (based on knowledge of the frequencies of initial events). Its purpose is to: identify all the pathways that lead to the main adverse event under certain circumstances; determining the minimum number of combinations of events that may lead to the main event; qualitative determination of the main causes of the adverse event; analysis of the sensitivity of individual events to deviations of system parameters.

A key theoretical basis for constructing FTA failure trees is the assumption that components in the ecosystem either work successfully or fail completely (Fig. 1). Failure tree is a deductive logical construction, which uses the concept of one final event (usually an accident or failure of an element of the system or the entire ecological and man-made system) in order to find all possible ways in which it can occur. A basic set of symbolic images can be used to graphically represent the simplest event tree.

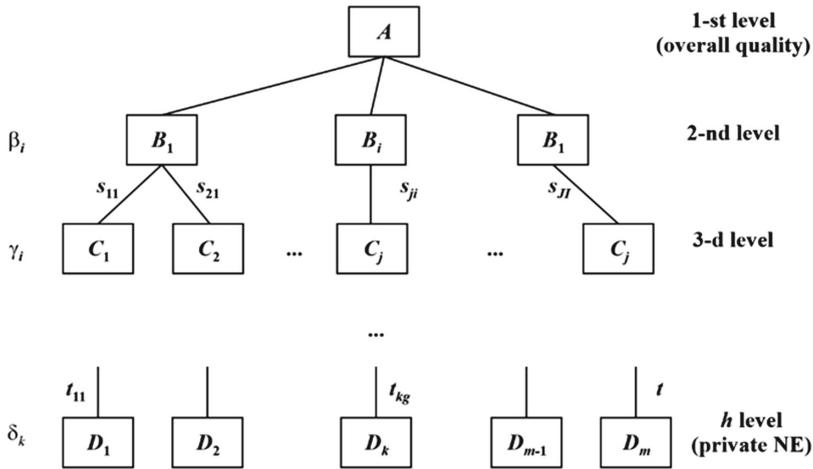


Fig. 1. Hierarchical structure of indicators of the object of monitoring

3.4 The Essence and Basic Principles of the Method of Analysis of Functional Stability of FME (C) A

The method of analysis of types, causes and consequences of failures and their criticality FME (C) A (failure modes and effect criticality analysis) is:

- formation of the hierarchy of the element-system and the definition of many elements, the failures of which are analyzed in terms of impact on the functional stability of the ecological-man-made system;
- determining the types of failures of each of the elements of the ecological and man-made system;
- analysis of the consequences of failures of each of them for the functional stability of the ecological and man-made system;
- determining the criticality of these failures as an integral characteristic, which includes, above all, the probability and severity of the consequences of failures (qualitative or quantitative rating scale);
- determination of the most critical failures based on the construction and analysis of the construction of two- (“probability” - “severity”), and in the general case - N-dimensional criticality matrix, in which each element is placed in a cell matrix according to its criticality.

Detection of failures, analysis of their causes and consequences, as well as their ranking according to the degree of criticality allows to solve the problem of optimal improvement of the functional stability of ecological and man-made systems. For example, this problem was solved by the authors when applying an integrated environmental assessment of the environment to analyze the environmental situation in the aerodrome areas of Ukraine in risk categories. Construction of an integrated risk assessment of chemical pollution (NS) was carried out in accordance with the hierarchical structure presented in Fig. 2. C_1, C_2, C_3, C_4, C_5 are the content of nitrogen dioxide, suspended solids (dust), sulfur dioxide, carbon monoxide, lead in the air respectively; C_6, C_7, C_8, C_9 are the content of manganese, iron, nitrates, nitrites in water respectively; $C_{10}, C_{11}, C_{12}, C_{13}, C_{14}$ are the content of copper, zinc, lead, cadmium, nickel in the soil respectively.

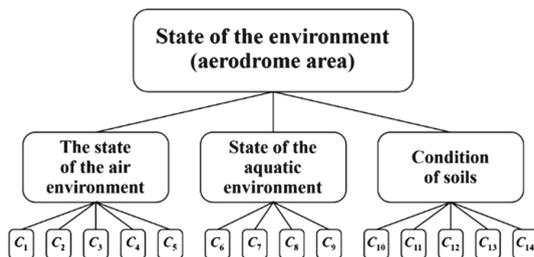


Fig. 2. Hierarchical structure for the implementation of integrated environmental assessment

A hierarchical approach to the analysis of the types of causes and consequences of failures and the analysis of FTA failure trees can be combined with the refinement procedure, which is the basis of the Event-B method. In this case, an abstract FMEA table and an abstract FTA failure tree are generated for the initial abstract Event-B model.

In the process of performing the detailing procedure, when the next transition from a more abstract to a more specific system model is performed, the detailing of the FMEA-table and the FTA failure tree is performed [20]. Thus, ultimately, we have a hierarchy of FMEA tables and FTA trees corresponding to the hierarchy of Event-B system models. In turn, the decomposition operation of the Event-B model will correspond to the operation of splitting a more abstract FMEA table into several next-level FMEA tables, as well as a more abstract FTA tree into several next-level FTA trees. At each stage of detailing, the criticality of failures is assessed and the least critical can be excluded from further analysis. This approach solves the problem of the size and complexity of FMEA and FTA analysis for multicomponent ecological systems, as well as to consider the additional property of traceability (tracing) of FMEA and FTA analysis, which is especially important for independent examination and verification.

3.5 FMECA-Analysis of the Event-B Model

The results of the analysis of the type, causes and consequences of failures should be systematized. For each of the failures detected as a result of FTA-analysis, the violated invariant of the Event-B model is determined, as well as possible reasons that need to be clarified taking into account the implementation of hardware and hardware, such as: damage to communication lines management; distortion of the management team during its transfer; failure of hardware (AS) control system (SU); failure of the AU SU or automation [18].

Thus, it should be concluded that it is necessary to systematize formal methods in order to select one or more to solve specific practical problems, their integration in the case when they complement each other and allow to obtain a synergistic effect from joint use. In any case, projects for critical applications that use formal development methods should be verified using expert judgment.

3.6 The Method of Alternating Parameter Changes (Gauss-Seidel Method) in the Environmental Management System

According to this method, the movement along each phase coordinate of the environmental management system occurs in turn. Initially, the movement along the first phase coordinate, and the remaining phase coordinates remain unchanged. This movement continues until the derivative of the control quality function at this coordinate is zero, i.e. From the last condition the first coordinate is defined. After that, the values of the first phase coordinate and the remaining coordinates, except the second, remain unchanged, and the second coordinate changes until the condition is met that the derivative of this coordinate is zero. From this condition the second phase coordinate is determined. Then the third coordinate

changes, etc. The search process continues until all partial derivatives are zero (until all derivatives are less than the sensitivity threshold of the system).

3.7 The Method of Random Blind Searches in the System of Ecological Management

The essence of the method is to find the extremum by randomly changing the first phase coordinate of the environmental management system. In the initial state of the system, the phase coordinates are given a random increment and the increment of the system quality function is determined. If it is negative (when searching for the maximum), the system returns to its original state and the next test step is performed. This is repeated until we get a positive increase. Then the system is transferred to this new state, from which new random steps are taken.

This method consists in the fact that from the initial state of the system several trial random steps are made and for each of them there is an increase in the quality function of the environmental management system. These increments, as components of the vector, determine the direction of the most intensive change in the quality function of the system. A work step is performed in this direction, and then the search cycle is repeated.

In some cases, it's best to combine different search methods. In particular, at the beginning of the search far from the extreme point, you can use the method of the fastest descent, and near the extreme to move to the gradient method. This process is illustrated in Fig. 3. It is assumed that the quality function determined on the plane, and that its graph has the shape of a bowl. The curves of the quality function are isoclines, i.e. areas in which the value of the quality function is constant. The arrows coming out of the points show the direction opposite to the gradient at that point. It is seen that the descent gradient leads us to the bottom of the bowl, i.e. to the point where the value of the quality function is minimal (minimum environmental hazard).

Gradient descent works in spaces with any number of dimensions. If the curvature of a given function is very different in different directions, the gradient descent may require many iterations to calculate the local minimum with the required accuracy in abnormal, emergency and catastrophic situations.

4 Computer Simulation, Results and Discussion

In the synthesis of management decisions in the system of environmental management, we consider the proposed consideration of the following formalization.

Man-caused environmental catastrophe - a major environmental accident (catastrophe) at a technical facility, which entails mass deaths and/or environmental disasters. One of the features of a man-made environmental catastrophe is its randomness (thus it differs from terrorist attacks) and is usually opposed to natural environmental disasters. In English, the term "man-made disaster" is rarely used. American and British authors in such cases usually speak of "technological catastrophes" and "technological disasters" (Industrial disasters), transport accidents

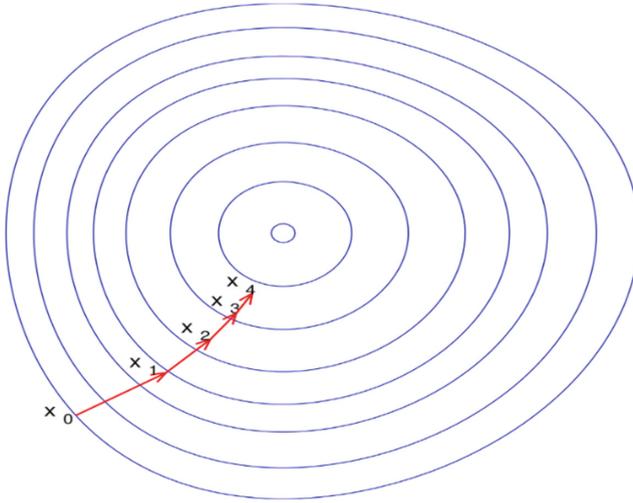


Fig. 3. Illustration of the method of eliminating environmental hazards

(Transportation disasters), and together with wars and terrorist attacks combine them into the concept of “man-made disasters”. “(Man-made disasters).

It should be borne in mind that an ecological catastrophe - an irreversible change in natural complexes associated with the mass death of living organisms. As a rule, a natural disaster is understood as an unexpected, terrible in its consequences for man disruption of the normal course of natural processes. According to most scientists, natural disasters are a normal and necessary for the self-development of natural systems geophysical process, which includes all sorts of rapid variations, deviations, disruptions, the surprise of which - the result of poor study of this natural phenomenon. Engineering and technical measures to prevent emergencies at the territorial, local and site levels should ensure the predominant use of active protection methods.

Accidents and catastrophes can be sources of man-caused emergencies:

- accident is a dangerous man-made accident that poses a threat to human life and health in an object, defined territory or water area, leads to the destruction of buildings, structures, equipment and vehicles, disruption of production or transport process, as well as damage to the environment the environment;
- catastrophe is a major accident with human casualties.

The concept of “risk” has many definitions. One of them defines risk as a possible danger and action at random in the hope of a happy outcome. Necessary elements of risk are danger, uncertainty, accident. Risk arises where there is a danger of uncertainty. Uncertainty not related to danger is not considered a risk. Danger, if foreseeable, is also not considered a risk. Environmental risk is the probability of negative changes in the environment, or long-term adverse effects

of these changes that occur due to environmental impact. Environmental risk is associated with possible disruption of the natural functioning of landscapes and ecosystems (deforestation, desertification, floods, villages, secondary soil salinization, earthquakes, etc.), sustainable development of economic systems (fires, explosions, accidents on power systems, nuclear power plants), chemical industry enterprises, transport accidents, etc.), when the change in their condition is accompanied by harmful consequences for humans. One of the components of environmental risk-uncertainty, i.e. lack of understanding of the phenomenon, its poor predictability or even complete unpredictability. Uncertainty arises due to the following reasons: lack of sufficient information, insufficient accuracy of data, lack of understanding of the phenomenon, randomness, organically inherent in the phenomenon. As an example, using the formal method, we will identify five typical scenarios of events due to an emergency caused by a fire in a radioactively contaminated forest.

Information on the location of the fire was obtained using aerospace technology (space image). After that the scheme of consecutive inspection of the area at planned inspection was defined: by parallel slogans; behind the converging spiral; on the divergent spiral (Fig. 4). Then the results of the survey with the help of unmanned aerial vehicles were put on the map of the survey area (Fig. 5) for the formation of management decisions.

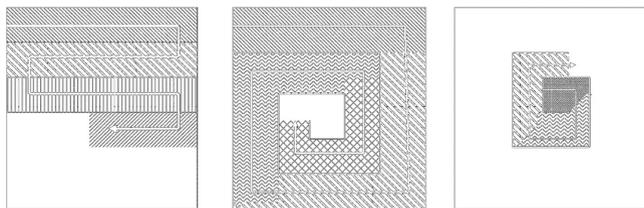


Fig. 4. Scheme of sequential survey of the area during the scheduled inspection: parallel slogans; behind the converging spiral; behind the divergent spiral

The first scenario is favorable when an extreme situation begins to develop, but the measures taken eliminate all preconditions for possible accidents and catastrophes in the shortest possible time, while the whole system (enterprise or natural object, in our case forest) returns to its original state. Where various processes of accumulation of everyday negative factors still take place.

This scenario may correspond to the second stage of forest fire development, when there is preheating and drying of combustible material with the release of water vapor, which forms the first radiation predictor of fire in radioactively contaminated forest - raising radioactive vapors from fires over tree canopies and forming contrasting in comparison with the general background of the radioactive point region - a weak pulsating radiation anomaly. It can be detected from manned and unmanned aerial vehicles and, accordingly, to ensure timely

of the anomaly will also depend on the time and intensity (speed) of combustion of combustible material, which determine the characteristics of convection flows. Such an anomaly can be detected in a timely manner from manned and unmanned aerial vehicles, which allows for prompt decision-making to attract the necessary forces and means to eliminate the developing fire.

The third version of the scenario of emergency development caused by a fire in a radioactively contaminated forest is called “balancing at the border”. This is sometimes spoken of as a miracle, as something fantastic, impossible. In fact, it happens that a set of measures taken, efforts, actions, etc., an accident or other catastrophic event can be avoided. Subsequent actions reduce the situation to the second and then to the first scenario. The result is a “balancing on the border” scenario with a positive result. This is the transformation of a stable radioactive anomaly into a bulk radioactive anomaly, in which the radioactivity increases significantly as it approaches from the periphery to the center. On its periphery there are local radioactive plumes, which are due to the movement of air masses. The intensity of radioactive radiation of the bulk anomaly increases, which is caused by an increase in combustion temperature. Such an anomaly can be reliably detected from manned and unmanned aerial vehicles and can be detected in time by spacecraft through radiation and infrared channels. Timely information allows management to decide on the organization and deployment of the necessary forces and means to eliminate the fire that begins in a radioactively contaminated forest.

The fourth scenario is the development of an emergency caused by a fire in a radioactively contaminated forest, when balancing on the border fails to contain the situation and a catastrophic event occurs - the forest fire enters an active phase that develops and rapidly covers all large areas. This scenario is called “balancing on the border” with a negative result. It can be matched to the fifth stage of forest fire development, when the flame burning begins with the release of smoke, carbon dioxide, water vapor and unburned gases, when the fourth radiation predictor of fire in a radioactively contaminated forest. There is a development (transformation) of bulk radioactive anomaly, the appearance in its structure of several epicenters - foci of the greatest radioactivity, as well as additional radioactive plumes, there is a significant increase in radioactivity of the anomaly. It is also confidently detected from manned and unmanned aerial vehicles and can be detected in a timely manner by spacecraft through radiation and infrared channels. The information obtained in this way allows management to decide on the organization and deployment of the necessary forces and means to eliminate the fire that begins in the radioactively contaminated forest, as well as the mobilization of additional forces and means to control the emergency.

The fifth variant of the scenario of emergency development caused by a fire in a radioactively contaminated forest is called inevitable, when an avalanche-like development of the event is such that an emergency is imminent. The time interval left before a catastrophic event can be used to minimize future consequences. For example, as a result of deliberate arson (sabotage) the fire covers a huge area in a short period of time. It is fixed by space not only on radiation and heat, but also optoelectronic (television) channels. There are not enough forces

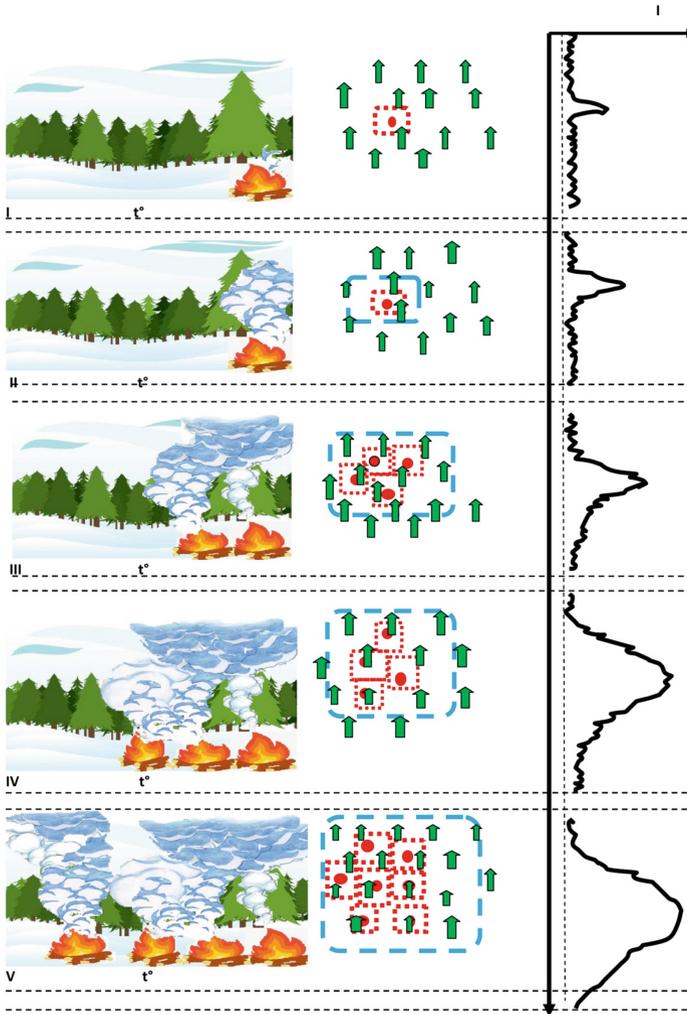


Fig. 6. Schemes of emergency scenarios

and means to put out the fire, so evacuation is needed first of all to save people. Schemes of the scenarios described above are presented in Fig. 6.

5 Conclusions

The paper proposes directions for the complex application of the formal Event-B method for the development of environmental management systems.

The practice of prevention and elimination of environmental emergencies indicates the general nature of risks in the field of environmental, industrial and

occupational safety. Accordingly, the management of these risks is optimally carried out on a single basis - in the framework of the creation of environmental management systems (management objects are considered man-made hazardous objects, ecosystem objects). The creation of such management systems involves the unification of approaches to ensuring activities in different directions.

The joint use of the formal Event-B method and failure analysis methods, primarily the analysis of types and consequences of critical failures FME (C) A and its modifications allows to expand the constructive use of formal methods, extending their capabilities to systems critical to failures, and physical defects, as well as defects of interaction (information and physical). Moreover, under certain conditions, such integration can be used to create so-called resilient systems - ecological systems that are resistant not only to failure but also to changes in requirements and parameters of the environment (systems capable of evolving in real time - real-time evolvable systems). In this case, the set of permissible events and their corresponding invariants expands. The main areas of further research and development in this area include: detailing options and procedures for integrating the Event-B method and methods of failure analysis (FME (C) A, HAZOP, FTA and their modifications); development of the assessment base - metrics of sets of invariants for calculation of indicators of reliability and guarantee capacity; creation of appropriate instrumental support that integrates the existing tools of formal development and analysis “Event-B” Rodin “platform, FME (C) A utility, etc.

References

1. Abrial, J.-R.: A system development process with event-b and the Rodin platform. In: Butler, M., Hinchey, M.G., Larrondo-Petrie, M.M. (eds.) ICFEM 2007. LNCS, vol. 4789, pp. 1–3. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76650-6_1
2. Abrial, J.: Formal Method Course. Zürich (2005)
3. Abrial, J.: Modeling in Event-B: System and Software Engineering. Cambridge University Press, Cambridge (2009)
4. Analysis Techniques for System Reliability - : Procedure for Failure Modes and Effects Analysis. (FMEA), IEC 60812 (2006)
5. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopolymers Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
6. Bowen, J.P.: Formal methods in safety-critical standards. In: Proceedings of the Software Engineering Standards Symposium (SESS 1993), Brighton, UK, pp. 168–177. IEEE Computer Society Press (1993). <https://doi.org/10.1109/sess.1993.263953>
7. Clark, E.M., Gramberg, O., Peled, D.: Verification of program models. Model Checking, Moscow (2002)
8. Devyanin, P.N., Leonova, M.A.: The techniques of formalization of OS Astra Linux Special Edition access control model using Event-B formal method for verification using Rodin and ProB. *Prikl. Diskr. Mat.* **52**, 83–96 (2021). <https://doi.org/10.17223/20710410/52/5>

9. Edmunds, A., Butler, M.: Linking event-b and concurrent object-oriented programs. *Electron. Not. Theor. Comput. Sci. (ENTCS)* **214**, 159–182 (2008). <https://doi.org/10.1016/j.entcs.2008.06.008>
10. Fatrell, R.T., Schafer, D.F., Schafer, L.I.: Software project management. In: *Achieving Optimal Quality at Minimum Cost*. Moscow, Williams (2004)
11. Fault tree analysis (FTA): IEC. 61025 (2006)
12. Kharchenko, V.S.: Guaranteeability and guarantee systems: elements of methodology. *Radio-electron. Comput. Syst.* **5**, 7–19 (2006)
13. Lecomte, T.: Formal methods in safety-critical railway systems. In: *10th Brazilian Symposium on Formal Methods, Ouro Preto, Brazil* (2007)
14. Mashkov, V.: New approach to system level self-diagnosis. In: *11th IEEE International Conference on Computer and Information Technology, CIT 2011*, pp. 579–584 (2011). <https://doi.org/10.1109/CIT.2011.12>
15. Mashkov, V., Mashkov, O.: Interpretation of diagnosis problem of system level self-diagnosis. *Math. Model. Comput.* **2**(1), 71–76 (2015)
16. Schneider, S.P.: *The B-Method: An Introduction*. Cornerstones of Computing series, Hampshire (2001)
17. Sterritt, R., Rouff, C.A., Rash, J.L., et al.: Self-properties in NASA missions. In: *4th Int. Workshop on System/Software Architectures (IWSSA 2005) at International Conference on Software Engineering Research and Practice (SERP 2005)*, Las-Vegas, Nevada (USA), pp. 66–72 (2005)
18. Tarasyuk, O., Gorbenko, A., Kharchenko, V.: Practical aspects of applying the Invariant-based approach to the formal system development and verification. In: *Monograph of System Dependability, 2, Dependability of Networks*, Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej (2010)
19. Tarasyuk, O.M., Horbenko, A.V., Kharchenko, B.C.: Complexation of formal methods of development and analysis of reliability of Event-B and FME (C), *A. Math. Mach. Syst.* **2**, 166–177 (2010)
20. Woodcock, J., Larsen, P., aBicarregui, J.: Formal methods: practice and experience. *ACM Comput. Surv.* **41**(4), 1–36 (2009). <https://doi.org/10.1145/1592434.1592436>



Ecology Objects Recognition by Optimized Inverse Filtration of Textured Background

Roman Kvyetnyy , Olga Sofina  , and Yuriy Bunyak 

Vinnitsia National Technical, Vinnitsia, Ukraine
rkvetny@sprava.net, olsofina@gmail.com

Abstract. Recognition and elimination of textured background is the first step in computer monitoring of environmental objects by intelligence systems. The approach to develop filters of textured images features is offered. The problem of texture recognition is formulated as inverse problem to modeling of texture dynamic. The modeling is based on the approximation of the texture surface by sum of principal components in the manner of eigen harmonics (EH) which frequencies are related with the texture structure. The approach has some advantages, since the EH have simple form and shift invariance is an important quality in the case of large landscapes modeling and filtering. The texture is considering as a response of a linear dynamic system to an excitation signal of a known shape. The dynamic system is characterized by a transient response. The inverse transient response allows to restore the excitation signal and, if it differs from the original within the specified limits, it's a sign of texture recognition. The EH approximation is using to find transient response and its inverse form, to optimize it in accordance with criterion of minimal surface.

Keywords: Eigen-harmonic decomposition · Texture dynamic modeling · Inverse filtration · Object recognition · Optimization

1 Introduction

The problem of tracking natural environment state using technical devices with cameras is important today. The flow of information received by devices is too large to be analyzed by human efforts. Processing is carried out using artificial intelligence systems (AIS). Their task is to recognize objects and their state. These tasks are solved by learning of the AIS. The learning process consist of development of algorithms which recognize patterns of objects or their fragments. Natural landscapes have many types of patterns and so the AIS are complicated for this purpose. Therefore, methods of efficient recognizing, especially for real time tracking and monitoring, are necessary. The convolutional network is the most effective for these problems solution. The convolution of an object image with filter of its feature gives the desirable result. We consider the problem

of feature filters design with application to recognizing objects in landscape environment.

Natural landscapes in environment look like quasi-uniform or chaotic textures. Therefore, recognition of environmental objects by artificial intelligence systems is possible after detecting the objects on a textured background. The problem is to define features which distinguish an image objects from their background. These are statistical, structural and dynamic properties.

The filters have to be simple and appropriate for implementation in convolutional network. Elimination of the textured background allows to recognize target objects - pollutions, mammals, peoples, etc. Filters should be invariant to spatial or temporal changes of filtered textured background, such as forests, waves of the water surface, fields with various kinds of plants. On this condition it will be effective in object recognizing with minimal number of errors.

The main contributions of this paper are summarized as follows:

1. The method of texture modeling and filtering to objects recognition against texture background was offered.
2. The texture filter manner is carried out from the task of optimal object filtration basing on the linear convolutional texture model.
3. The abstract texture model with invariance to linear shift was presented. Its eigen decomposition was made in the harmonic basis.
4. The method of filter design was developed on the base of eigen harmonic decomposition of target texture.
5. As the filter design is ill-posed problem it was offered the optimization of the filter shape with regularization.
6. Two dimensional algorithm of the EHD decomposition which gives needed functions of the decomposition was considered.
7. The experiment of modeling and filtering of some typical for ecological problems natural textures by designed and known methods was made. The error level estimates have shown advantages of the offered method comparing to the known ones.

The rest of the paper is structured in the following way. The known solutions of the problem of textures modeling and filtering from the point of view of their implementation to dynamic textures modeling with properties of space and temporal invariance and possibility of real time implementation were investigated in Sect. 2. Based on the results of the analysis of related works, the problem of designing the filters with properties of invariance by using harmonic functions was formulated in Sect. 3. The texture model and objects filter are designed in the Sect. 4. The experimental analysis of the method is presented in the Sect. 5. The results of the experiment and some properties of the designed method are discussed in the Sect. 6. Section “Conclusions” summarizes the research results and indicates the direction for future work.

2 Literature Review

The latest methods of textures analysis are examined in [20, 25, 26, 33]. We will consider the methods from the point of view of their invariance to spatial and

temporal changes of target textures, also, possibility of their real time implementation. Most methods are based on statistical analysis, spectral transformations, and dynamic models of textured image signals. Our main interest is related with model based methods of image analysis.

The statistical methods are based on parametric and nonparametric models of the image textures such as Bayes analysis, Hidden Markov Models, Regression, Linear and Nonlinear Autoregression [3, 16, 20, 25–27, 33, 36]. The statistical nonparametric methods primary are nonlinear and not always give the textured image pattern estimates that are invariant in respect to spatial transforms of the image.

The most widely used correlation method is the Method of Co-occurrence Matrices [25, 33]. Further development of the correlation method includes definition of a nonlinear metric that relates to geometry of textures [6, 8, 27, 28].

The newest methods of object recognition are based on image signal transforms in the spatial-frequency domain, where spectral parameters of the image have simple statistic in a narrow range [17, 26]. These methods combine statistical models and the Fourier transform, and use a set of functions of a certain type. The most popular are wavelet functions and their generalizations for 2D transforms [12], classic Gabor functions [13, 23, 30].

The wavelet spectrum represents local geometric features of textures with a minimal number of significant coefficients; however, it does not represent the texture periodicity because wavelets have shift-variable spectrum. This problem was partially overcome by the introduction of the Complex Wavelet Transform [13]. Shift and rotate invariance have been investigated in some articles on object recognition [10, 21].

A desire to present textures with a minimal number of parameters has driven the development of new transforms, many of which are a combination of existing transforms. For example, the Radon transform in collaboration with common and wavelet based Fourier transforms, provides the possibility to present images in a frequency-angle domain that has a much lower dimension as it is affine-invariant [18, 35].

Exact texture shape description is not necessary in many applications. When texture is dynamic the spatial and temporal changes are more important than local surface pattern characteristics.

Combining the Eigen Vector Decomposition (EVD) of texture kernel with filters that resonant to certain image eigen modes gives most informative texture description [1]. This approach was developed with using methods of Independent Component Analysis, Principal Component Analysis, Empirical Mode Decomposition (EMD) and Singular Values Decomposition (SVD) [4, 22, 31, 34].

The decomposition methods make a projection of a textured image into a subspace configured by the most significant parameters. Usually the decompositions are used for full image transformation that complicates their application in real time systems.

In the case of dynamic textures are used dynamic models based on discrete differential equations [9, 15, 29]. The model synthesis is computationally compli-

cated. But the model implementation is not complicated and can be executed in real time using a common processor or arithmetic array [9].

The considered above methods are predominantly shift variant and do not allow real time implementation.

Based on the results of the analysis of related works, we can make the conclusion that the approach based on the approximation by eigen harmonics [19] gives needed solution of the problem. This solution is basing on the following:

- correlation matrix and other integral matrices such as Wigner integral are used for objects description because they represent the dynamic properties of large image in compact form;
- the asymptotic correlation matrix of a stationary process has Toeplitz structure and its eigen vectors are harmonic in nature are shift invariant [19];
- a textured image can be considered as a 2D stationary process with the Block Toeplitz correlation matrix.

So, the harmonic decomposition is a natural approximation of EVD or EMD. The parameters of harmonics can agree with periods of a texture structure and gray scale spatial variations. Such harmonics are considered as eigen or resonance with respect to texture structure. Also, the Eigen Harmonic Decomposition (EHD), based on planar and spherical functions, have been used for compact presentation of textured images, as well as human facial images [14, 24].

3 Problem Statement

The brief survey of textured image analysis methods allows to make the following conclusions:

- Dynamic model based methods of texture analysis and synthesis are the most effective. These methods can be used for analysis of static, dynamic, and quasi-regular textures. Models may be created with using a fragment of the image. The remainder of the image of an arbitrary size can be restored by the model operator using a generation process.
- An implementation of the structured object recognition using inverse in respect to object model schema yields to make recognition of the simple process that generate the texture by its model.
- The dynamic models are usually synthesized using the Fourier transform or decompositions mentioned above. The survey of previous research shows that the use of transforms and decompositions depends on the image type - full format, or only a typical fragment (known as texture kernel or as texton) [5]. When image full format is using the wavelet and Radon transforms combined with Fourier transform give appropriate results. When the texton is using the wavelet transforms and SVD ignore the structural information because they are shift-variable.
- In contrast to the mentioned decompositions, the EHD has simple analytical definition and join capabilities to approximation and interpolation of 2D fields [2]. Therefore, the EHD is an appropriate function basis for the design of a linear inverse filter.

So, we will consider the problem of object recognition on textured background as the linear inverse filtration which suppresses textured object structure and transforms it into simple for recognizing signal. At the same time, it leaves foreign objects signal fluctuations without change of their energy.

4 The Method of Inverse Filtration Based on the Eigen Harmonic Decomposition

4.1 Object Recognition on Textured Background as Problem of Optimal Filtration

The problem of objects recognition can be formulated as follows. Let in the space Ω is given a textured image X which includes an foreign object $O : O \subset \Omega$. The integral on the subset $\forall \Omega_O \subset \Omega : M(\Omega_O) < M(O)$ ($M(O)$ is size of region O) gives the texture model

$$\int_{\Omega_O} H(x, y; u, v)E(u, v)dxdy = X(u, v) + \varsigma_x : u, v \notin O \quad (1)$$

where $H(x, y; u, v)$ is the transient response and $E(u, v)$ is the excitation signal, ς_x - model error. The integral

$$\int_{\Omega_O} \tilde{H}(x, y; u, v)X(u, v)dxdy = \begin{cases} E(u, v) + \varsigma_E : u, v \notin O; \\ \varsigma_O \gg \varsigma_E : u, v \in O \end{cases} \quad (2)$$

reconstructs the excitation signal with error ς_E if its range of integration does not belong the object. Otherwise the error significantly exceeds ς_E . This condition is possible if errors ς_x and ς_E are minimal, that can be presented by the following functional.

$$I = \arg \min_{H, \tilde{H}} \left\{ \frac{1}{2} \int_{\Omega/O} \left((H \circ E - X)^2 + (E - \tilde{H} \circ X)^2 \right) dudv \right\} \quad (3)$$

where “ \circ ” denotes convolution. Euler-Lagrange variation of the functional (3) with account expressions (1) and (2) relatively parameters of the texture model H and object filter \tilde{H} gives an optimal schema of objects recognition.

$$(H \circ E - X) \circ E + (E - \tilde{H} \circ X) \circ X \approx 0. \quad (4)$$

The δ -function or Heaviside step function, uncorrelated white noise can be used as the excitation signal. Then up to a constant factor $H \circ E \circ E \propto H \circ \delta = H$. As a result, we obtain the following equation for optimal object filter and model operators characteristics.

$$H - \tilde{H} \circ R_{xx} \cong 0, \quad (5)$$

where $R_{xx} = X \circ X$ is the correlation matrix of the image signal. Since $H \circ \tilde{H} \cong I$, where I is the trivial operator. The convolution (5) with \tilde{H} yields the expression:

$$\tilde{H} \circ \tilde{H} \circ R_{xx} = (\tilde{H} \circ X) \circ (\tilde{H} \circ X) \cong I, \quad (6)$$

that means that filter \tilde{H} eliminates signal X components which form correlative relations. It is in inverse resonance with image texture shape. So, we define \tilde{H} as Inverse Resonance Filter (IRF). It is inverse to well-known Wiener filter [32].

The problem is to define principal components of an image texture which can be eliminated by simple linear convolution operation. The main demand to IRF properties is invariance to object space location and initial phase of its texture shape. This demand can be formulated as shift invariance. It points on choice of a method for retrieval of principal components of the object model. The shift invariance is inherent to harmonic functions.

4.2 Eigen Harmonic Decomposition of Textured Image

An ideal textured image of size $N \cdot P \times M \cdot Q$ can be represented as a tensor product of two matrices,

$$D = T \otimes B, \quad (7)$$

where matrix T of size $N \times M$ includes unit elements with insignificant fluctuations and matrix B of size $P \times Q$ is the texture kernel. Matrix (7) ranks equal to rank of the kernel matrix. Therefore it can be represented by using P left vectors and Q right vectors of the EVD. The Discrete Fourier Transform (DFT) in the basis of exponential functions [19] of the matrix T can be presented as:

$$T = F_N \text{diag}[10 \dots 0] F_M^H, \quad (8)$$

where $F_N = 1/N [z_i^k]_{i,k=0 \dots N-1}$, $z_i^k = \exp(-t2\pi ik/N)$, is the DFT operator whose size is pointed by low indexes. The kernel matrix has the following spectral decomposition:

$$B = F_p A_{FB} F_Q^H, \quad (9)$$

where A_{FB} is the spectral matrix of kernel B in the basis F . The substitution of expressions (8) and (9) into (7) yields the following decomposition of the textured matrix

$$\begin{aligned} D &= (F_N \otimes F_Q) (\text{diag}[1 0 \dots 0] \otimes A_{FB}) (F_Q^H \otimes F_M^H) \\ &= F_{NP} \text{diag}[A_{FB} 0 \dots 0] F_{MQ}^H \end{aligned} \quad (10)$$

In expression (10) it was used associativity of the tensor product and that $F_{NP} = F_N \otimes F_P$, $F_{MQ} = F_M \otimes F_Q$. It is possible when the basis is exactly aligned with texture structure.

The basis of EHD can be aligned with texture structure using the condition of shift invariance. The shift of textured matrix (7) initial point causes the cyclical shift of the kernel matrices. The shifted matrix relates with the original matrix B by simple expression

$$B^{t,\tau} = C_P^t B C_Q^{\tau H}, \quad (11)$$

where $B^{t,\tau}$ is the shifted matrix along t rows and τ columns, C_P is the cyclical shift operator of size $P \times P$ [7].

$$C_P^t = F_P \text{diag}[z_1^t \dots z_P^t] F_P^H; \quad C_Q^\tau = F_Q \text{diag}[z_1^\tau \dots z_Q^\tau] F_Q^H. \quad (12)$$

The texture matrix (7) with shifted kernel matrices (11) can be represented with account of expressions (9)–(12) as

$$\begin{aligned} D^{t,\tau} &= T \otimes B^{t,\tau} \\ &= F_{NP} \text{diag} [\text{diag} [z_1^t \dots z_P^t] A_{FB} \text{diag} [z_1^{\tau*} \dots z_Q^{\tau*}] 0 \dots 0] F_{MQ}^H, \end{aligned} \quad (13)$$

where “ $*$ ” is a complex conjugate. The spectral factorization of the textured image matrix (7) and (13) shows that the power spectrum of the kernel matrix is invariant to the cyclical transforms because the spectrum matrix of the shifted kernel (11) differs from the original by phase multiplies.

The cyclical shift of the ideal texture (7) can be changed by linear shift along coordinates which is natural for real texture. Such shift can be defined on the base of linear prediction approach and it has the manner of the Frobenius matrix:

$$K_P = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_P & -a_{P-1} & \dots & -a_1 \end{bmatrix} \quad (14)$$

with characteristic polynomial

$$1 + \sum_{i=1}^P a_i z^i = 0. \quad (15)$$

The polynomial (15) has the roots $z_{i=1\dots P}$, $z_i = \exp(t2\pi f_i)$, f_i - eigen or resonance frequencies. The matrix (14) can be defined as the Linear Shift Operator (LSO). Its spectral factorization for coordinates OX and OY is the following.

$$K_{xP}^t = Z_{xP} \text{diag} [z_{x1}^t \dots z_{xP}^t] Z_{xP}^\sharp; \quad K_{yQ}^\tau = Z_{yQ} \text{diag} [z_{y1}^\tau \dots z_{yQ}^\tau] Z_{yQ}^\sharp, \quad (16)$$

where $Z_{xP} = [z_{xi}^t]_{i=1\dots P}^{t=0\dots P-1}$, $Z_{yQ} = [z_{yi}^\tau]_{i=1\dots Q}^{t=0\dots Q-1}$, “ \sharp ” is the pseudoinversion. The linearly shifted texture kernel matrix can be presented by the EHD (16) in the following manner,

$$\begin{aligned} B^{t,\tau} &= K_P^t B K_Q^{\tau T} \\ &= Z_{xP} \text{diag} [z_{x1}^t \dots z_{xP}^t] Z_{xP}^\sharp Z_{xP} A_{ZB} Z_{yQ}^T Z_{yQ}^{\sharp T} \text{diag} [z_{y1}^\tau \dots z_{yQ}^\tau] Z_{yQ}^T \\ &= Z_{xP} \text{diag} [z_{x1}^t \dots z_{xP}^t] A_{ZB} \text{diag} [z_{y1}^\tau \dots z_{yQ}^\tau] Z_{yQ}^T, \end{aligned} \quad (17)$$

where “ T ” is the transposition, and A_{ZB} is the spectral matrix of the matrix B in the basis Z .

$$A_{ZB} = \tilde{Z}_{xP}^\sharp B \tilde{Z}_{yQ}^{\sharp T}. \quad (18)$$

Expression (17) shows the spectrum invariance to linear shift operations that is equivalent to expression (13). The kernel periodicity in the textured image is defined by the condition

$$B^{mP,nQ} = B, \quad (19)$$

where m and n are the arbitrary integer values. This condition allows presentation of the full image D by overdetermined matrices $\tilde{Z}_{xP} = [z_{xi}^t]_{i=1\dots P}^{t=0\dots NP-1}$, $\tilde{Z}_{yQ} = [z_{yi}^t]_{i=1\dots Q}^{t=0\dots MQ-1}$ using the spectral matrix (18).

$$D \approx \tilde{Z}_{xP} A_{ZB} \tilde{Z}_{yQ}^T. \tag{20}$$

Expression (20) is analogous to (10) but it is approximate because the pseudoinverse of the overdetermined matrices yields approximate results.

Linear shift is natural for many models of textured images, both static and dynamic. As it follows from condition (19) the shift operations do not change the texture signal energy. Therefore, the shift operator should be unitary. The operator (14) is unitary if its characteristic polynomial (15) roots lie on the unit circle in the complex domain.

4.3 Development of IRF for Texture Recognition

The inverse schema (2) means restoration of the originating signal, the matrix $E = [e_{i,k}]_{i=0\dots M-1}^{k=0\dots N-1}$, using measurement data $D = [d_{i,k}]_{i=0\dots M-1}^{k=0\dots N-1}$ of size $M \times N$ under the assumption that mentioned matrices are connected by linear operator (1) with unknown transient characteristic $H = [h_{m,n}]_{m=0\dots P-1}^{n=0\dots Q-1}$. The measured data matrix includes samples of additive noise $\xi_{i,k}$ associated with texture model error. Formally the model can be written in the discrete manner as the convolution equation

$$d_{i,k} = \sum_{m=0}^{P-1} \sum_{n=0}^{Q-1} h_{m,n} e_{i-m,k-n} + \xi_{i,k}, \tag{21}$$

where $d_{i,k} \in \Xi : M(\Xi) = M(\Omega_O)$ is a base region of the texture image, free from foreign objects, 2D Heaviside step function

$$e_{ik} = \begin{cases} 1, (i \geq 0^+) \wedge (k \geq 0^+); \\ 0.5, i = 0, k = 0; \\ 0, (i \leq 0^-) \vee (k \leq 0^-) \end{cases} \tag{22}$$

is the generating process in expression (21). The textured image is specified in the range $(i \geq 0^+) \wedge (k \geq 0^+)$ and is a response of linear dynamic system (21) on the excitation (22).

Let us to consider the problem of generating signal reconstruction using an inverse filter with transient characteristic $\tilde{H} = [\tilde{h}_{m,n}]_{m=0\dots P-1}^{n=0\dots Q-1}$. The generating signal has constant value E in the range of image, so the problem appears in the following way:

$$\sum_{m=0}^{P-1} \sum_{n=0}^{Q-1} \tilde{h}_{i+m,k+n} d_{m,n} = E + \varsigma_{i,k}, \tag{23}$$

where $\varsigma_{i,j}$ represents the error noise. The dispersion value of the noise in (23) serves as a criterion for filter matching with object or the texture.

We can compile the system of Eqs. (23) for shifted data matrices

$$\sum_{m=0}^{P-1} \sum_{n=0}^{Q-1} \tilde{h}_{i+m, k+n} d_{m+t, n+\tau} \cong E : \forall d_{m+t, n+\tau} \notin O \subset \Omega, \quad (24)$$

and find minimum square solution of the system (24) with respect to transient characteristic $\tilde{h}_{m,n}$ elements by pseudoinversion of the extended data matrix. But such solution will be not shift invariant. It obtains the shift invariance in shift invariant functions basis. Let the elements of the image matrix in (23) are presented by EHD as the following series:

$$d_{i,k} = \sum_{m=1}^P \sum_{n=1}^Q A_{m,n} z_{xm}^i z_{yn}^k + \varepsilon_{i,k}, \quad (25)$$

where $A_{m,n}$ represents the amplitudes of the two-dimensional harmonic functions $z_{xm}^i z_{yn}^k$, $\varepsilon_{i,k}$ is the approximation error, which has much less dispersion than signal power. The main parameters of harmonic functions $z_{xm}^i = \exp(t2\pi f_{xm}i)$ and $z_{yn}^k = \exp(t2\pi f_{yn}k)$ in (25) are values of resonance frequency f_{xm} , f_{yn} .

It is not necessary to use a full textured image for series (25) parameter determination. Some base region $\Xi \geq B$ of the size $n_x \times n_y : n_x > 2P; n_y > 2Q$ is sufficient. This region should cover the texture kernel B . The EHD of the base region is equivalent to (9) and it can be represented in the matrix form as (18) and (20). The EHD of transient characteristic

$$h_{i,k} \cong \sum_{m=0}^{P-1} \sum_{n=0}^{Q-1} H_{m,n} z_{xm}^i z_{yn}^k. \quad (26)$$

The substitution Eqs. (25), (26) into (21) with neglecting the error yields the following spectrum of the IRF.

$$\tilde{H}_{m,n} \cong H_{m,n}^{-1} = E_{m,n} A_{m,n}^{-1}. \quad (27)$$

where:

$$E_{m,n} = \sum_{i=0}^{P-1} \sum_{k=0}^{Q-1} e_{i,k} z_{xm}^i z_{yn}^k = \sum_{i=0}^{P-1} \sum_{k=0}^{Q-1} z_{xm}^i z_{yn}^k.$$

The transient characteristic of the IRF (23) for texture recognition can be defined using equation

$$\tilde{H} = \left[\tilde{h}_{i,k} = \sum_{m=1}^P \sum_{n=1}^Q E_{m,n} A_{m,n}^{-1} z_{xm}^i z_{yn}^k \right] i = 0 \dots P - 1; k = 0 \dots Q - 1. \quad (28)$$

The schema (26)–(28) implements the approach to filter development basing on Fourier transform. It differs from well-known classic schema by using EHD instead usual discrete Fourier transform. The EHD allows avoid of small amplitudes $A_{m,n}$ influence because all eigen fluctuations have comparative value levels.

4.4 Optimization of the IRF

In the given above schema the base Eqs. (25), (26) and (27) are approximate. So, this approach gives an estimate of the aim transient characteristic. It needs in optimization in accordance with a criterion of quality. The total variance or function norm can serve as criterion of the optimization. The function \tilde{H} surface area

$$\sum_{\tilde{H}} = \int_{\Xi} \sqrt{1 + \tilde{H}_x^2 + \tilde{H}_y^2} dx dy, \tag{29}$$

where $H_x = \partial H(x, y) / \partial x$ and $H_y = \partial H(x, y) / \partial y$, provides minimization of fluctuations and serves as main part of objects characteristic features.

The optimization must ensure the fulfillment of the condition

$$\tilde{H} \circ H \cong I \tag{30}$$

with account of surface (29) minimization. The problem (30) can be presented by the functional of the IRF spectrum (27) optimization.

$$I_{\tilde{h}} = \arg \min_{\tilde{H}} \left\{ \sum_{i,k \in \Xi} \left\| \sum_{m=1}^P \sum_{n=1}^Q \sum_{t=0}^{n_x-P} \sum_{\tau=0}^{n_y-Q} \tilde{H}_{m,n} z_{xm}^{i+t} z_{yn}^{k+\tau} h_{t,\tau} - \delta_{i,i_0} \delta_{k,k_0} \right\|^2 + \lambda \cdot \text{Reg}(\tilde{H}) \right\}$$

where (i_0, k_0) is a central point of the texture base region, λ is the parameter of regularization,

$$\text{Reg}(\tilde{H}) = \sum_{i,k \in O} \sqrt{1 + \left(\sum_{m,n} \tilde{H}_{m,n} dz_{xm} z_{xm}^i z_{yn}^k \right)^2 + \left(\sum_{m,n} \tilde{H}_{m,n} dz_{ym} z_{xm}^i z_{yn}^k \right)^2}$$

where $dz_{x(y)m} = 0.5(z_{x(y)m} - z_{x(y)m}^{-1})$ are spectrum coefficients introduced by discrete derivatives of the harmonic functions. Euler-Lagrange derivative of the functional is the next.

$$\begin{aligned} \frac{\partial I_{\tilde{h}}}{\partial \tilde{H}_{u,v}} = & \sum_{m,n} \tilde{H}_{m,n} \left(2 \sum_{i,k \in O} \sum_{t,\tau} h_{t,\tau} \sum_{i,k \in O} z_{xm}^{i+t} z_{yn}^{k+\tau} \sum_{t',\tau'} h_{t',\tau'} z_{xu}^{i+t'} z_{yu}^{k+\tau'} \right. \\ & \left. - \lambda (dz_{xm} dz_{xu} + dz_{yn} dz_{yv}) \sum_{i,k \in O} z_{xm}^i z_{yn}^k z_{xu}^i z_{yv}^k \text{Reg}(\tilde{H})^{-1} \right) \tag{31} \\ & - \sum_{i,k \in O} \sum_{t',\tau'} h_{t',\tau'} z_{xu}^{i+t'} z_{yu}^{k+\tau'} = 0 \end{aligned}$$

The PQ expressions (31) create the system of equations for iterative evaluation of the IRF transient characteristic optimal spectrum \tilde{H} elements. The

initial spectrum is $\tilde{H}_{m,n}^{(0)} = \tilde{H}_{m,n}$ in (27). The regularization parameter λ can be chosen by the help of the condition of convergence of the q first iterations: $\sum_{u,v=0} \left\| \tilde{H}_{u,v}^{(t+1)} - \tilde{H}_{u,v}^{(t)} \right\|^2 \theta \leq \sum_{u,v} \left\| \tilde{H}_{u,v}^{(t)} - \tilde{H}_{u,v}^{(t-1)} \right\|^2$, where $\tilde{H}_{m,n}^q$ is the result of q -th iteration step, $t = 1, 2, \dots, q$, θ - a positive value. The iteration process can be stopped if $\sum_{u,v} \left\| \tilde{H}_{u,v}^{(t+1)} - \tilde{H}_{u,v}^{(t)} \right\|^2 \leq \varepsilon_{\tilde{H}}$, where $\varepsilon_{\tilde{H}}$ is a small value.

4.5 Eigen Harmonic Decomposition for the IRF Design

The first step of the IRF design is the EHD of the target textured object template using base region Ξ . Let's consider a two-dimensional EHD that ensures the fulfillment of the condition of invariance to shift - the eigenvalues lie on the unit circle of the complex domain.

The simplest way is solution of the eigenvalue problem of the LSO (14) for two space coordinates. This approach gives appropriate result if the 2D signal is uniform. If the signal is not uniform, the full matrix of 2D data should be used.

The eigenvalues problem may be written for 2D data $d_{m,n}$ of size $M \times N$ as matrix pencils [11]:

$$D_x^{(2)} - z_x D^{(2)}; D_y^{(2)} - z_y D^{(2)}, \quad (32)$$

where $D_{x(y)}^{(2)} = \left[D_{i+k+x}^{(y)} \right]_{i=0 \dots M-L-1}^{k=0 \dots L-1}$, $D_{i+k+x}^{(y)} = [d_{i+k+x, m+n+y}]_{n=0 \dots N-L-1}^{m=0 \dots L-1}$, z is the spectral parameter, and L is the splitting parameter. If the 2D data include $P/2$ real harmonics and do not contain noise and value $P \leq L$ the rank of matrix pencils (32) is equal to P , the data can be presented by P vectors of the SVD. If z_x and z_y are equal to the matrix pencils eigenvalues, then the expressions (32) are equal to zero. In order to solve the eigenvalues problem the following matrices can be used relatively to x and y coordinates.

$$Z_{Ex}^{(2)} = \left(D^{(2)} \right)^\# D_x^{(2)}, Z_{Ey}^{(2)} = \left(D^{(2)} \right)^\# D_y^{(2)}. \quad (33)$$

The SVD can be used for the operation of pseudoinversion in (33) by using of the largest singular values and corresponding them vectors. As we mentioned above, the texture image dynamic model needs in unitary shift transform by the matrices in the left part (33) as well as by LSO (14). This condition may be supported by including into (33) direct and backward shift transforms to eliminate influence of the dumping factors of the spectral parameters in the case of data matrices splitting. The data matrices in (32) can be accomplished by inverse direction components as $[D^{(2)} J D^{(2)}]$, where J is the matrix with unit cross-diagonal. The SVD of the correlation matrix

$$R^{(2)} = \left[D^{(2)} J D^{(2)} \right]^T \left[D^{(2)} J D^{(2)} \right].$$

is the following:

$$R^{(2)} = U \cdot \text{diag}[\mathbf{s}] \cdot V^T, \quad (34)$$

where U and V are the matrices of the left and right orthogonal vectors, \mathbf{s} is the vector of singular values. Matrices (33) may be rewritten as

$$Z_{Ex}^{(2)} = (U_0^H U_0)^{-1} U_0^H U_x, (U_0^H U_0)^{-1} U_0^H U_y \quad (35)$$

using expression (34), where matrices U_0, U_x, U_y have P vectors which correspond P largest singular values in decomposition (34). The matrix U_0 is the basis matrix of expression (35), the shifted matrices U_x, U_y have structure that depends on the direction of the shift toward x and toward y . The basis matrix U_0 is formed by extracting from matrix U the last $M - L - 1$ rows and each $N - L - 1$ rows starting from the last row. The matrix U_x is formed by extracting from matrix U in (34) the first $M - L - 1$ rows. The matrix U_y is formed by each $N - L - 1$ row of matrix U , starting from the first row. The eigenvalues z_{xi}, z_{yi} ($i = 1 \dots P$) can be defined as eigenvalues of the matrices (35).

5 Experiment

The IRF (23) generates an image $G : \tilde{d}_{i,k}^{(t)} \in G; t = 1, 2, 3$ by filtration of the texture image $D : d_{i,k}^{(t)} \in D$. The foreign objects can be recognized by the following logical filter:

$$\text{if } \bigcup_{t=1}^3 \left(\left| \tilde{d}_{i,k}^{(t)} - E \right| > 3\sigma_\zeta^{(t)} \right) \text{ then } g_{i,k}^{(t)} = d_{i,k}^{(t)} \text{ else } g_{i,k}^{(t)} = 0; t = 1, 2, 3, \quad (36)$$

where:

$$\sigma_\zeta^2 = \frac{1}{n_x n_y} \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \left(\tilde{d}_{i,j} - E \right)^2$$

is the dispersion of the filtration error. The top indices in the brackets point on color components.

The initial and filtered by nonlinear autoregressive method [26, 36] and IRF images with logical and statistical analysis in accordance with (36) are presented in Fig. 1, 2, 3 and 4, also, it was used the know statistical method based on co-occurrence matrices [26, 27], its results are not shown because they are similar to autoregression method. The images present the examples of solution of the problems of mammals and people tracking in natural landscapes, pollution control. The quality of the recognition was estimated by percent of false and lost pixels, the estimates are shown in Table 1.

The base regions Ξ of size $n_x \times n_y = 80 \times 80$ for EHD models synthesis were chosen in upper left corner of each initial image. The order of the EHD models $P \times Q$ was $12 \times 12 \dots 18 \times 18$. The criterion of the order choosing is the error power level of the EHD approximation (25) of the template region - it should be not higher -40 dB in respect to template image power. This is the main limitation on the application of this approach to modeling and filtering textured images.

Table 1. Recognition quality estimated by percent of false and lost pixels

	Percent of false and lost pixels (%)			
Type of method	Fig. 1,	Fig. 2	Fig. 3	Fig. 4
IRF	0.25	1.04	17.73	0.68
Autoregression	5.59	100	46.40	4.78
Co-occurrence	6.12	82.04	52.11	7.32



Fig. 1. Broadband camera marine surface filtration and wale recognition by autoregression and IRF methods



Fig. 2. Spectral camera marine surface filtration and underwater wale recognition by autoregression and IRF methods



Fig. 3. Marine surface filtration and pollution recognition by autoregression and IRF methods



Fig. 4. Filtration of sandy desert surface with dunes and object recognition by autoregression and IRF methods

The examples have shown, that the model (21) and its spectral presentation by EHD (25) are successfully approximate quasi regular textures in Fig. 1, 3 and chaotic textures in Figs. 2, 4. At the same time the IRF is sensitive to small changes in image signal dynamic, for example underwater wale in Fig. 1 and man in Fig. 4 it was successfully tracked. In the last case, the object has slight differences from the texture in the form of sand dunes. Spectral cameras are often used in video measurement processes because they have more high sensitivity in comparison with usual wideband cameras, but they give images that are difficult to perceive because of their randomness, nevertheless, the IRF made it possible to highlight the faint object.

The most difficult texture is on Fig. 2. The background is dynamical with high amplitude of variability and it is not uniform. The figures have shown that the IRF can be used for processing of wide diapason of textures and it has advantages in respect to known methods.

6 Results and Discussion

Analysis of methods of textured image modeling has shown that current researches focus on the decompositions on components and integral transforms that give the most informative parameters and reflect the spatial and temporal properties of the texture. These methods are founded on the symmetry of a texture in respect to shift and rotate transforms. The symmetry means that exist character of texture pattern transform which reflects its changes in space or time. This type of dynamic transform can be represented by the linear shift operator that models step-type changes of texture pattern as well as its periodicity. The eigenvectors of the LSO can serve as a basis for eigen-harmonic decomposition of the texture with invariance to shift transforms.

We have shown in (6) that if the inverse filtration is resonance in respect to texture pattern than such filter is optimal. The term “resonance” has many means. In general, this is equivalence of a target object to some templates of pattern. The resonance with harmonic fluctuations is mostly known.

The next step of the filtration optimization is evaluation of the optimal transient characteristic (31) in spectral domain. Although the EHD basis provides a mathematically correct solution of the inverse problem with a certain number of estimated harmonic principal components, the elements of the amplitude matrix in (27) and their inverse values are sensitive to empirical data chaotic fluctuations. Therefore, the optimization of the found spectrum (27) needs for enforcing of main harmonic components. This is achieved by the help of additional restrictions on the form of the inverse transient characteristic in (29), (30). The restriction of minimal surface allows select principal spectral components. It was needed up to 5 iterations (31) to achieve the convergence condition with $\varepsilon_{\tilde{H}} = 1..e - 8$.

The IRF can be used in AIS of convolutional type for object feature extraction. Each target object in the manner of quasi-uniform texture can be presented by single filter.

7 Conclusions

The new approach to textured objects recognition technique was presented. It is basing on the EHD of the target texture template and approximation of it in the EHD basis. The given spectrum is using to design IRF. The filter is inverse to transient characteristic of the linear convolutional model of the texture. The IRF allows to recover texture excitation signal which has simple known form. Two-dimensional EHD of quasi regular image signal is developed. The filter is optimized to eliminate small fluctuation caused by solution of the ill-posed problem. The shift-invariance properties of the EHD basis are inherited by the filter, that allows to create a model and eliminate quasi-chaotic dynamic textures and detect objects against their background. The filter has shown in The experiments showed advantages of IRF comparatively to the known methods of autoregression and co-occurrence matrices which are shift invariant too.

The considered above method of textured image filtration and foreign objects recognition differs from other known method by following features: each texture is characterized by unique own filter; filter implementation is feasible in real time using graphic processor units; regulation of resolution and quality is provided by filter order change; is based on the principal eigen harmonic decomposition that is more convenient than the EMD or the EVD because it has the analytical form and may be adapted to any image or image sample fragment.

The further perspective of the proposed method is to extend ability of the IRF to recognize not only textured images but objects too. Complex natural objects can be segmented on some typical fragments that can be considered as textures. An object can be recognized after recognition of its fragments.

References

1. Ade, F.: Characterization of texture by “eigenfilter”. *Signal Process.* **5**, 451–457 (1983). <https://doi.org/10.5244/C.18.66>
2. Bunyak, Y.A.: Harmonic analysis of wave fields. *J. Commun. Technol. Electron.* **43**(3), 239–243 (1998)
3. Celeux, G., Forbes, F., Peyrard, N.: EM procedures using meanfield-like approximations for Markov model-based image segmentation. *Pattern Recogn.* **36**(1), 131–144 (2003). [https://doi.org/10.1016/S0031-3203\(02\)00027-4](https://doi.org/10.1016/S0031-3203(02)00027-4)
4. Xiong, C., Xu, J., Zou, J., Qi, D.: Texture classification based on EMD and FFT. *J. Zhejiang University-SCIENCE A* **7**(9), 1516–1521 (2006). <https://doi.org/10.1631/jzus.2006.A1516>
5. Charalampidis, D.: Texture synthesis: textons revisited. In: *IEEE Trans. Image Process.* **15**(3), 777–787 (2006). <https://doi.org/10.1109/tip.2005.860604>
6. Clarenz, U., Diewald, U., M., R.: Processing textured surfaces via anisotropic geometric diffusion. *IEEE Trans. Image Process.* **13**(2), 248–261 (2004). <https://doi.org/10.1109/TIP.2003.819863>
7. Elliot, D.F., Rao, R.: *Fast Transforms: Algorithms, Analysis and Applications*. Academic Press, New York (1983)
8. Fu, Y., Huang, T.S.: Image classification using correlation tensor analysis. *IEEE Trans. Image Process.* **17**(2), 226–234 (2008). <https://doi.org/10.1109/tip.2007.914203>

9. Fujita, K., Nayar, S.K.: Recognition of dynamic textures using impulse responses of state variables. *Pattern Recogn.* **31**, 1496–1509 (2003)
10. Guo, Z., Zhang, L., Zhang, D.: Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recogn.* **43**(3), 706–719 (2010). <https://doi.org/10.1016/j.patcog.2009.08.017>
11. Hua, Y.: Estimating two-dimensional frequencies by matrix enhancement and matrix pencil. *IEEE Trans. SP* **40**(9), 2267–2280 (1992). <https://doi.org/10.1109/78.157226>
12. Huang, K., Aviyente, S.: Information-theoretic wavelet packet subband selection for texture classification. *Signal Process.* **86**, 1410–1420 (2006). <https://doi.org/10.1016/j.sigpro.2005.07.032>
13. Kingsbury, N.G.: Complex wavelets for shift invariant analysis and filtering of signals. *J. Appl. Comput. Harmon. Anal.* **10**(3), 234–253 (2001). <https://doi.org/10.1006/acha.2000.0343>
14. Kvyetnyy, R., Bunyak, Y., Sofina, O., Romaniuk, R.S., Tuleshova, A.: Blur recognition using second fundamental form of image surface. In: *SPIE*, vol. 9816. id. 98161A 9 (2015). <https://doi.org/10.1117/12.2229103>
15. Kvyetnyy, R., Sofina, O., Orlyk, P., Wójcik, W., Orazalieva, S.: Improving the quality perception of digital images using modified method of the eye aberration correction. In: *SPIE - The International Society for Optical Engineering*, vol. 10031, p. 1003113 (2016). <https://doi.org/10.1117/12.2249164>
16. Lazebnik, S., Schmid, C., Ponce, J.: Affine-invariant local descriptors and neighborhood statistics for texture recognition. In: *ICCV*, vol. 6, pp. 221–229 (2003). <https://doi.org/10.1109/ICCV.2003.1238409>
17. Li, H., Liu, G., Zhang, Z.: A new texture generation method based on pseudo-DCT coefficients. *IEEE Trans. on Image Processing* **15**(5), 1300–1312 (2006). <https://doi.org/10.1109/TIP.2005.863970>
18. Liu, G., Lin, Z., Yong, Y.: Radon representation-based feature descriptor for texture classification. *IEEE Trans. Image Process.* **18**, 921–928 (2009)
19. Marple, S.L.: *Digital Spectral Analysis with Applications*. Prentice-Hall Inc, Englewood Cliffs, New Jersey (1987)
20. Mehta, R., Egiazarian, K.: Dominant rotated local binary patterns (DRLBP) for texture classification. *Pattern Recogn. Lett.* **71**, 16–22 (2016). <https://doi.org/10.1016/j.patrec.2015.11.019>
21. Nguyen, T.P., Vu, N.S., Manzanera, A.: Statistical binary patterns for rotational invariant texture classification. *Neurocomputing* **173**, 1565–1577 (2016)
22. Nunes, J.C., Niang, O., Bouaoune, Y., Delechelle, E., Bunel, P.: Bidimensional empirical mode decomposition modified for texture analysis. In: *13th Scandinavian Conference, SCIA*, pp. 171–177 (2003). <https://doi.org/10.5555/1763974.1764003>
23. Palm, C., Lehmann, T.V.: Classification of color textures by Gabor filtering. *Graph. Vision* **11**(2), 195–219 (2002). <https://doi.org/10.5555/tip.762282.762288>
24. Pattichis, M.S., Bovik, A.C.: Analyzing Image Structure by Multidimensional Frequency Modulation. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 1–36 (2006). <https://doi.org/10.1109/TPAMI.2007.1051>
25. Radke, R. J. and Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: a systematic survey. *IEEE Trans. Image Process.* **14**(3), 294–307 (2005). <https://doi.org/10.1109/TIP.2004.838698>
26. Randen, T., Husoy, J.H.: Filtering for texture classification. A comparative study. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(4), 291–310 (1998). <https://doi.org/10.1109/34.761261>

27. Rezaei, M., Fekri-Ershad, S.: Texture classification approach based on combination of random threshold vector technique and co-occurrence matrixes. In: ICCSNT, vol. 4, pp. 2303–2306 (2011). <https://doi.org/10.1109/ICCSNT.2011.6182434>
28. Sayadi, M., Sakrani, S., Fnaiech, F., Cheriet, M.: Gray-level texture characterization based on a new adaptive nonlinear auto-regressive filter. *Electron. Lett. Comput. Vis. Image Anal.* **7**(1), 40–53 (2008). <https://doi.org/10.5565/rev/elcvia.158>
29. Soatto, S., Doretto, G., Wu, Y.N.: Dynamic textures. *Int. J. Comput. Vision* **51**, 91–109 (2003). <https://doi.org/10.1023/A:1021669406132>
30. Vyas, V.S., Rege, P.: Automated texture analysis with gabor filter. *GVIP J.* **6**(1), 35–41 (2006). <https://doi.org/10.1.1.94.2430>
31. Wall, M.E., Rechtsteiner, A., Rocha, L.M.: Singular value decomposition and principal component analysis. *A Practical Approach to Microarray Data Analysis*. Norwell, MA (2003). https://doi.org/10.1007/0-306-47815-3_5
32. Wiener, N.: *Time Series*. MIT Press, Cambridge (1964)
33. Xie, X.: A review of recent advances in surface defect detection using texture analysis techniques. *Electron. Lett. Comput. Vis. Image Anal.* **7**(3), 1–22 (2008)
34. Yang, Z., Yang, L.: Decomposition based on classical empirical mode decomposition and radon transform. In: *IMECS 2009*, pp. 18–20 (2009)
35. Yarman, C.E., Yaz, B.: An Inversion Method for the Exponential Radon Transform Based on the Harmonic Analysis of the Euclidean Motion Group. In: *SPIE 2006*, vol. 1, pp. 6142–61424A (2006). <https://doi.org/10.1117/12.647807>
36. Zhang X., W.X.: Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation. *IEEE Trans. Image Process.* **16**(7), 887–896 (2008). <https://doi.org/10.1109/TIP.2008.924279>

Theoretical and Applied Aspects of Decision-Making Systems



Information Technology to Assess the Enterprises' Readiness for Innovative Transformations Using Markov Chains

Marharyta Sharko¹, Olha Liubchuk¹, Galina Krapivina¹ ,
Natalia Petrushenko² , Olga Gonchar³ , Kateryna Vorobyova⁴,
and Nataliia Vasylenko⁵ 

- ¹ State Higher Educational Institution “Pryazovskyi State Technical University”,
Dnipro, Ukraine
lyubchuk@ukr.net
- ² Ukrainian Academy of Printing, Lviv, Ukraine
natalia.velikaya@gmail.com
- ³ Khmelnytsky National University, Khmelnytsky, Ukraine
o.i.gonchar@i.ua
- ⁴ Limkokwing University of Creative Technology, Selangor, Malaysia
- ⁵ Kherson State Agrarian and Economic University, Kherson, Ukraine
neve80@ukr.net

Abstract. The paper presents the results of research regarding the processes of assessing the degree of readiness of enterprises for innovative activities, functioning in conditions of uncertainty of economic ties and relations. A conceptual model of an innovative activity management system has been developed, aimed at improving the diagnostic and decision-making mechanisms based on the use of Markov chain tools. For its practical implementation, a simulation model has been created for assessing the degree of readiness of enterprises for innovation in the form of a directed graph, in which the vertices represent the states of the process, and the edges represent transitions between them. A distinctive feature of the model is that it is not time, but the sequence of states and the number of a step with a hierarchy of sampling intervals, which is considered as an argument on which the process of assessing the degree of readiness of enterprises for innovative activities depends. The flexibility of such a model is ensured by adaptability to the influences of the external environment with the possibility of adjusting it to each information situation.

Keywords: Innovative activities · Markov chains · Enterprises · Information control systems

1 Introduction

One of the areas of successful functioning of enterprises in the context of dynamic changes in the external environment is their willingness to innovate. The ability of enterprises to carry out innovative activities is manifested in the ability

to obtain certain changes in the mechanism of enterprise functioning in order to achieve the final result - the release of competitive products. The innovative activity of enterprises is a synthetic property that accumulates heterogeneous factors of exogenous and endogenous nature. Management of innovative activity in conditions of uncertainty of the influence of the external environment is basically based on the creation of systemically related methods for studying the current and predicted state of production by synthesizing information obtained using both a synergistic approach and mathematical, including probabilistic methods.

Currently, innovative activities are carried out not only under conditions of risk, but also under conditions of uncertainty, characterized by the globalization of economic processes, the complication of interactions between market actors and the influence of the external environment. The choice of the dominance of alternatives for managing innovation and assessing the degree of readiness of enterprises for its implementation become the prerogative of intuition.

The process of assessing the degree of readiness of enterprises for innovative activities carries elements of randomness associated with the fact that even the current state of production at the initial moment of time is determined through the initial probability distribution.

The choice of the first step to change the current situation is associated with a quantitative assessment and adjustment of one of the factors of production activity, which leads to a shift in the starting point of the process of establishing the degree of readiness of an enterprise for innovative transformations. After completing the operations related to the adjustment of the subsequent factor, the starting point will again shift towards reducing the process of introducing innovative activities, etc. Thus, the process of changing the position of the reference point is of a random nature, characterized by an arbitrary choice of the corrected factor with discrete time characteristics of the duration of the first and subsequent steps and a countable set of states. Such a process will be Markov, since subsequent states of the starting point of the process of innovative transformations do not depend on past states.

The problem of quantitatively assessing the ability of enterprises to carry out specific types of innovative activities and the degree of readiness of enterprises to innovative transformations is always relevant.

The aim of the research is to develop information systems and technologies for determining the degree of readiness of enterprises for innovative activities based on the organization and regulation of innovative processes in developing production using Markov chains.

2 Related Works

Innovation can affect any aspect of an enterprise. Dynamic changes in the market conditions of management require enterprises to implement continuous technological and strategic changes to maintain and preserve existing positions. At the same time, strategic decisions should take into account the development of the

market and the external environment to a greater extent than internal factors. In general, an enterprise that implements innovative activities must have an external orientation towards the consumer market.

In [3,25,26], innovation management of enterprises is applied to study the characteristics of innovation and assess business strategies. In [23], the model of Markov process is applied to study the economic efficiency of a firm. The accuracy of determining the parameters of the development of economic systems is estimated. In [10], the theory of Markov chains is used to determine the vectors of supply and demand states. The intensity matrix is determined by transforming the logarithmic function of the corresponding transition matrices to polynomials.

In [24], the issues of constructing stochastic models of economic systems based on Markov processes with discrete states and with discrete time are considered.

In [4], the issues of modeling the economic sustainability of enterprises using Markov chains are considered, where economic sustainability is understood as the properties of enterprises during a certain time to achieve the goals of functioning and development. In [12], the Markov model is used to calculate the probabilities of transitions between states of patients as a set with deviations in the anatomy of the lymphatic drainage system. Demonstrated the ability to quantitatively assess the risk of disease. In [15], the basic principles of a homogeneous Markov network with a fixed number of states and a discontinuous period are considered.

Discrete-time Markov chains are used in [14,21,22] to simulate the impact of the COVID-19 pandemic on the five main sectors of the Kenyan economy that contribute significantly to GDP growth. The simulation results enable global investors to understand various aspects of economic stimulus planning to mitigate the impact of the economic recession.

In [8], the problem of testing the asymptotic distribution of transition probabilities of the Markov sequence of a parametric family is considered; in [6], the results of modeling the Monte Carlo method based on Markov chains when creating records belonging to the same object obtained from data belonging to different sources are described. An information-entropy model of the base for making managerial decisions under conditions of uncertainty is presented in [17]. Mathematical support for eliminating the influence of the human factor on navigation equipment systems under conditions of uncertainty and risk is presented in [1,2,11,13]. The use of a Markov chain model to assess the trajectory of medical students' progress is presented in [27], in [28] for short-term forecasting of wind energy and power production by trade winds at various spatial and temporal scales in [7] for segmentation of medical images in [5] for short-term demand forecasting on the water.

As follows from the above review, the practical applications of Markov chains are numerous and varied. Selected fragments of the presented experience were used in the development of the methodology of these studies. Methods for assessing the degree of readiness of enterprises for innovation require the improvement of optimization and probabilistic techniques.

3 Materials and Methods

With regard to the problem of determining the degree of readiness of enterprises for innovative activities in conditions of uncertainty of the influence of the external environment, the parameters of organizational activity in industrial production can be used as research material, and the Markov chain can be used as methods.

A Markov chain is a method for modeling random events, which is a discrete sequence of phases, each of which is located in discrete state spaces. Changes in the state of the production system in the Markov chain are transitions, and the corresponding probabilities associated with changes in states are transition probabilities.

The main idea of the Markov chain is that there is only one current state and therefore one transition to one subsequent state. The main Markov property is that at any moment of time the conditional distribution of future states with given current and past states depends only on the current state, but not on past states. Thus, a Markov chain is a sequence of random events with a finite number of transitions, implemented in practice with discrete time and discrete state space. The original probability distribution is:

$$P(x_0 = S) = q_0(S) \forall S \in E \tag{1}$$

where: S is the discrete states; q_0 is the probability distribution at a point in time $t = 0$.

The set E is a finite number of possible states.

$$E = \{e_1, e_2, \dots, e_n\} \tag{2}$$

Range of values of a random variable $\{x_n\}$ is the state space, and the value n is the step number. The probabilities of transition from one state to another are represented in the form of square matrices:

$$P_{ij}(n) = P(x_{n+1} = j | x_n = i) \tag{3}$$

$$S_1 S_2 S_n$$

$$\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_n \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{bmatrix} \tag{4}$$

The elements, p_{ij} denote the probability of transition from the state S_i to the next. The matrix of transition probabilities expresses the probability that a state at time $n + 1$ is subsequent for other states.

$$P(x_{n+1} = S_n + 1 | x_n = S_n) = P(S_n, S_{n+1}) \forall (S_n, S_{n+1}) \leq E * E \tag{5}$$

A Markov chain will be homogeneous if the matrix of transition probabilities does not depend on the step number:

$$P_{ij}(n) = P_{ij} \tag{6}$$

To pass from an initial state to a subsequent one, it is necessary to determine the probability of this transition in n steps. According to the Kolmogorov-Chapman equation, the matrix of transition probabilities in n steps in a homogeneous Markov chain is the n -th power of the matrix of transition probabilities in one step:

$$P(x_n = S_n | x_0 = S_0) = P^n \tag{7}$$

The fact of the use of Markov chains to determine the degree of readiness of enterprises for innovative activities lies in the fulfillment of the following provisions. The production system be in states S_1, S_2, \dots, S_n . Transitions during the implementation of innovations are possible only at points in time which characterize the stages of implementation, i.e. are considered steps. The argument of the Markov chain is the step number.

Let us to denote by $S_i^{(k)}$ the event that after k steps the production system will be in the state S_i . Events $S_1^{(k)}, S_2^{(k)}, \dots, S_n^{(k)}$, form a complete group of incompatible events. The process of introducing innovations can be represented as a chain of events $S_1^{(0)}, S_2^{(1)}, S_3^{(2)}, \dots, S_n^{(k)}$. If for each step the probability of transition from any state S to a state does not depend on when and how the system came to state S_i , then such a sequence of events will be a Markov chain. Let's take a closer look at this process. The production system be in any of the states of innovation implementation S_1, S_2, \dots, S_n , i.e. one event after another from a complete group of events can occur. Denote the probabilities of these events:

$$\begin{bmatrix} P_1(1) = P(S_2^{(1)}) & P_2(1) = P(S_2^{(1)}) & P_n(1) = P(S_n^{(1)}) \\ P_1(2) = P(S_1^{(2)}) & P_2(2) = P(S_2^{(2)}) & P_1(2) = P(S_n^{(2)}) \\ \dots & \dots & \dots \\ P_1(k) = P(S_1^{(k)}) & P_2(k) = P(S_2^{(k)}) & P_n(k) = P(S_n^{(k)}) \end{bmatrix} \tag{8}$$

For each step number, the condition is satisfied:

$$P_1^{(k)} + P_2^{(k)} + \dots + P_n^{(k)} = 1 \tag{9}$$

It is necessary to find the probabilities of the states of the production system for any step k . When assessing the degree of readiness of enterprises for innovative activities, innovation processes in the production system can be depicted as a sequence of discrete states. For any step in the innovation process, there are Markov chain transition probabilities. Transitional probabilities can be written in terms of conditional probabilities:

$$P_{ij} = P(S_j^{(k)} / S_j^{(k-i)}) \tag{10}$$

A Markov chain at any moment of time can be characterized by vectors by the row C_i of the matrix of transition probabilities P . If we multiply the vector row, describing the probability distribution at a certain stage of the implementation of innovations, by the matrix of transition probabilities, we obtain the probability distribution at the next stage of implementation. With regard to determining the degree of readiness of enterprises to innovate under conditions of uncertainty of the influence of the external environment, characterized by elements of randomness, the purpose of Markov chains is to search for such a combination of characteristics and parameters of innovation activity that would allow improving the mechanisms of diagnostics and decision-making in a visualized form at various levels implementation of innovations.

The readiness of enterprises to innovate can be represented as the internal capacity of production for possible transformations. If the transition of the system from one state to another occurs at pre-fixed times with the accumulation of the corresponding volume of investment resources, then we have a serial Markov process in discrete time. If the transition is possible at any random moment of time, we have a Markov process with continuous time.

In the process of introducing innovations, when assessing the degree of readiness of an enterprise for innovative transformations, events B_1, B_2, \dots, B_i the probabilities of which are known from the existing experience in the implementation of similar situations, may occur. These events characterize the parameters of innovative transformations in the implementation of innovations. The onset of events B_i transfers the enterprise to one of the discrete states S_1, S_2, \dots, S_m . A hit to any state is considered a random event with discrete time.

Each combination of parameters describing the informational situation of the state of the production system at the time of implementation of innovations is assigned a certain probability, which is written as a row of the matrix of states. In this case, the sum of the probabilities in the rows of the matrix is equal to one. All possible states of the parameters of innovation processes, which characterize the degree of readiness of enterprises for innovation, are sorted out. These iterations are carried out for various states of innovation activity with their own parameters and probabilities. This provides the possibility of constructing stochastic models of production systems based on Markov random processes with discrete states and discrete time. Since the processes of innovative transformations do not have a continuous time stamp, we use the stages of implementation of innovations as the time, which characterize the sequential approximation of the degree of readiness of enterprises for innovative activities to achieve economic results.

The distribution of probabilities in assessing the degree of readiness of enterprises for innovation does not depend on time, but depends only on the transitions from the current state to the corresponding stages of innovation implementation. Developing the proposed methodological approach, it is possible to establish the dependence of the degree of readiness of enterprises for innovative activities on the initial current state of the enterprise, creating the necessary basis for managing innovation.

A conceptual model of the degree of readiness of enterprises for innovative activities, compiled on the basis of an analysis of the basic principles of using Markov chains, is presented in Fig. 1.

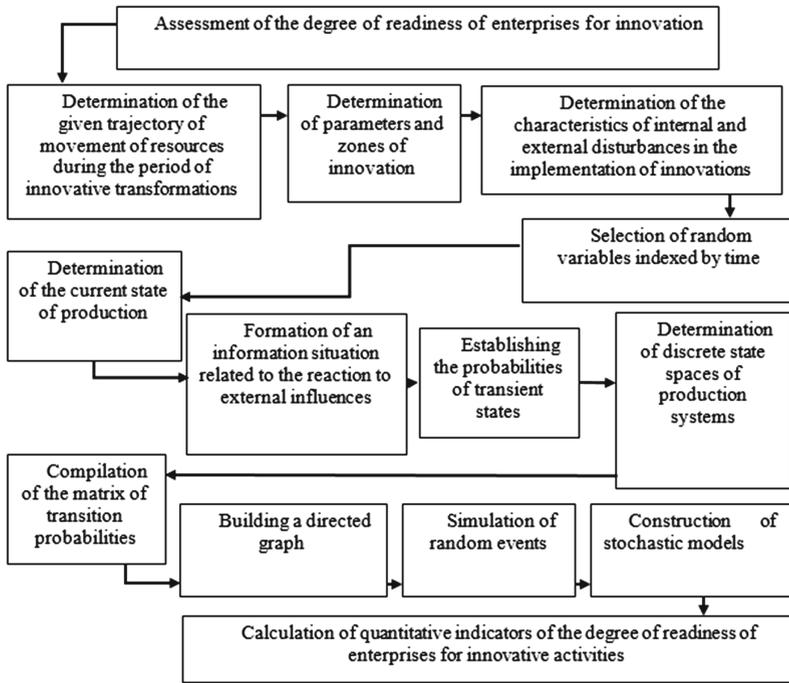


Fig. 1. Conceptual model of the enterprise innovation management system

Presented in Fig. 1 formalization of the main actions to assess the degree of readiness of enterprises for innovation, allows establishing the relationship between the individual characteristics of the management system of innovation [19]. This is the basis for predicting the competition of enterprises before and after the introduction of innovations [18,20]. A feature of the presented model is to take into account the multidirectional interests of participants in innovative transformations associated with the possibility of obtaining additional profit and with a constant share of risk present in this process. When making fundamentally different decisions either on the introduction of innovations or on their rejection, only an assessment of the current situation of production activity will be unmistakable. Many factors influence the results of innovation implementation. So the probability of getting an advantage can be 0.25 while the probability of risk is 0.75. The key concepts of the Markov chain in the implementation of innovations are shown in Fig. 2.

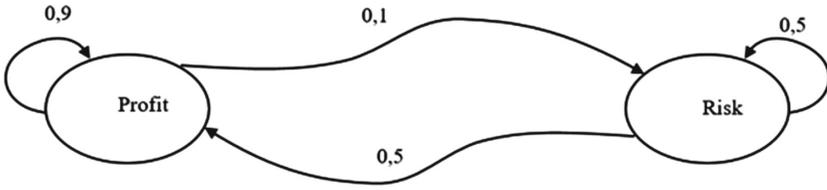


Fig. 2. Key concepts of the Markov chain in the implementation of innovations

In the presented scheme, the Markov chain represents a set of transitions, with its own probability distribution, satisfying the Markov property. If the Markov chain has N possible states, then the matrix will have the form $N \times N$ and each row will have its own probability distribution. The distribution of probabilities with three possible states of the degree of readiness of enterprises for innovative activities is shown in Fig. 3.

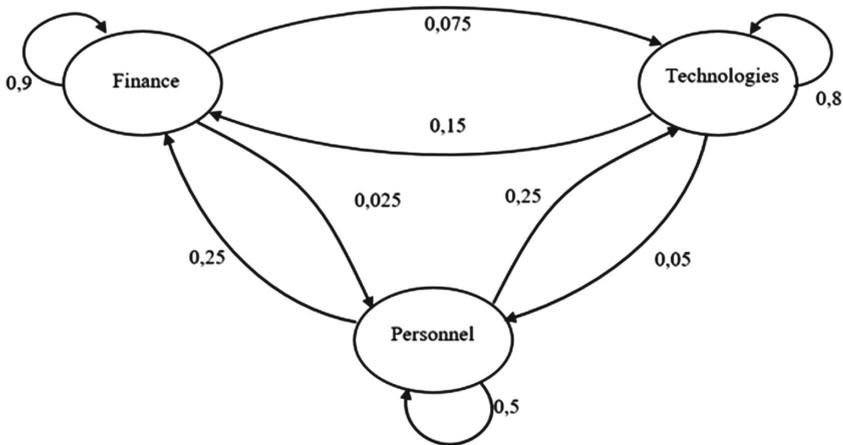


Fig. 3. Probability graph of a Markov chain with three possible states

A random stochastic process for determining the degree of readiness of an enterprise for innovative transformations is a set of random variables indexed by a set T , which denotes different moments of initiation of innovative activity in production.

The first step in creating a Markov chain to determine the degree of readiness of enterprises for innovation is the formation of a transition matrix. The current state of the production system is the initial state, while the remaining states are subsequent. The probabilities of the state chosen for the present in the subsequent time interval will be determined by one of the indicated situations, without their simultaneity. Having connected in this way all the probabilities of

the transition of these states, we obtain the matrices of the transition probabilities of the Markov chain. Each transition from state i to state j is characterized by the transition probability P_{ij} , which shows how often, after entering the i -th state, it makes a transition to the j -th state. Of course, such transitions occur randomly, but if you measure the frequency of transitions over a sufficiently long time, then it will coincide with the probability of transitions. When assessing the degree of readiness of enterprises to introduce innovations, various mutually exclusive situations are possible:

- the situation of seeking funding sources;
- the situation of the availability of the necessary equipment, in the absence of the appropriate competence of personnel to use it;
- the situation of the availability of the competence of personnel and the corresponding equipment, in the absence of binding to the solution of specific practical problems;
- a situation of lack of methodological support and forced caution, based on the search for possible reasons for postponing the introduction of innovations.

Economic decisions for assessing the degree of readiness of enterprises for innovative activity, taking into account many uncertain factors within the framework of the theory of Markov chains, postulate the choice of the best alternative, for which the apparatus of the theory of probability can be used. Markov chains allow you to generate events.

4 Experiment, Results and Discussion

To implement the proposed conceptual model for managing innovation in conditions of uncertainty, a simulation model has been created in the form of a directed graph, in which the vertices represent the states of the process, and the edges represent transitions between them. The simulation model allows you to record the current characteristics of production activities at any time. The flexibility of such a model is ensured by its adaptability to environmental influences. The key factors of such models are input variables due to responses to external influences.

Since the graph is oriented, it is not possible to get to another from every state. Each connection has its own probability. The transition from one node to another can pass through any random period of time. It is proposed as an argument, on which the process depends, to consider not the time t , but the number of the step $1, 2, \dots, n$. A random process in this case will be characterized by a sequence of states. If the initial probability distribution and the matrix of transition probabilities are known, then the total probabilistic dynamics of the process is determined and it can be calculated cyclically.

As a practical example of constructing a digraph of a Markov chain, reflecting the management of innovative activities in conditions of uncertainty, we selected the states in which the priority of the location of the parameters is characterized by the most significant characteristics: finance, equipment, personnel competence, methodology for processing research results. The transitions between

these states are selected as edges or arcs of the digraph components. Weights of arcs, expressed in terms of empirically determined probabilities, compiled on the basis of a generalization of information published in [9,16] are presented in Table 1.

Table 1. The current state of the production system

No	Parameters	Weights of arcs	Vertices of graphs
1	Financing	0.2	V1
2	Equipment	0.6	V2
3	Competence	0.1	V3
4	Methodology	0.1	V4

Large-scale technological changes, the science-intensive technology implementation, digitalization, intelligent and information networks, along with the complexity and dynamism of market processes lead to uncertainty in the functioning of enterprises. It is of interest not only to fix the factors that hinder the development of innovation, but also the possibility of developing strategic directions and methods of their decisions.

The innovations in information management systems and technologies implementation for assessing the degree of readiness of enterprises for innovative activities using Markov chains requires the creation of appropriate innovation potential, which depends on many economic factors. In the conditions of dynamic changes and uncertainty of influence of external environment the basic determining factors of introduction of innovations are financing, the equipment, competence and readiness for innovations. Having chosen the availability of financing as the main parameter of the degree of readiness for innovative activity, the subsequent states can be represented in the form of a 4×4 matrix. The values of the probabilities of the parameters for assessing the degree of readiness of enterprises for innovation at different stages of implementation are presented in Table 2.

According to tables Table 1 and Table 2, the production system can be in one of four states.

Table 2. The current state of the production system

Next state	Financing	Equipment	Competence	Methodology
Current State				
Financing	0.2	0.6	0.1	0.1
Equipment	0.1	0.6	0.1	0.2
Competence	0.1	0.7	0.1	0.1
Methodology	0.2	0.1	0.6	0.1

If enterprises operate steadily in a stable mode, the probability of the enterprise's readiness for innovation according to Table 2 if funding is available 0.2. To determine the degree of readiness of enterprises to innovate, it is necessary to purchase the appropriate equipment. This probability will be 0.6. The probability that the acquisition of the necessary competencies is sufficient to determine the degree of readiness of enterprises for innovation activity 0.1. The probability that the availability of the results methodology is sufficient to determine the degree of readiness of enterprises for innovation activity.

If the operating conditions of the enterprise are unstable with predictable seasonal expected changes controlled by insurance funds, then the probability of the enterprise's readiness to innovate only through the organization of financing is 0.1. The probability of determining the degree of readiness of enterprises for innovation in these conditions through the purchase of equipment is 0.6. The probability of having the required competencies is 0.1 and the corresponding methodology is 0.2.

If the working conditions of the enterprise are subject to sharp indecent fluctuations caused by changes in the pressure of the external environment, then the probability that the enterprise's readiness to innovate only through the organization of financing will be equal to 0.1. The probability of determining the degree of readiness of the enterprise through the purchase of equipment 0.7. The probability of having the required competencies is 0.1 and the corresponding methodology is also 0.1.

If enterprises operate in conditions of uncertainty in the pressure of the external environment, then the probability that the enterprise's readiness to innovate only through the organization of financing will be 0.2. The probability of determining the degree of readiness of the enterprise through the purchase of equipment is 0.1. The probability of having the required competencies is 0.6 and the corresponding methodology is also 0.1.

In the initial state of production according to table. 1 the probabilities of the current state of the production system by providing funding 0.2, equipment 0.6, competence 0.1 and methodology 0.1. It is necessary to determine what is the probability of the readiness of enterprises for innovative activities in the subsequent stages of the priority of using the parameters under various options for the pressure of the external environment.

$$p(0) = (0.2, 0, 4, 0.1, 0.1) \quad (11)$$

$$T = \begin{bmatrix} 0.2 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.1 & 0.2 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.6 & 0.1 \end{bmatrix} \quad (12)$$

If we multiply the row vector describing the probability distribution at a given stage of time by the matrix of transition probabilities, then we obtain the probability distribution at the next stage of time. The probability that in one step the system will pass from the state S_0 to the state S_1 and the operating conditions of the enterprise will be stable.

$$P(1) = (0.2, 0.6, 0.1, 0.1) \times \begin{bmatrix} 0.2 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.1 & 0.2 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.6 & 0.1 \end{bmatrix} = (0.13, 0.56, 0.15, 0.16) \quad (13)$$

The probability that being in state S_1 , in one step, the system will go to state S_2 under unstable but predictable changes in environmental conditions is equal to:

$$P(2) = (0.13, 0.56, 0.15, 0.16) \times \begin{bmatrix} 0.2 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.1 & 0.2 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.6 & 0.1 \end{bmatrix} \quad (14)$$

$$= (0.129, 0.535, 0.18, 0.156)$$

Under the condition of instability of the enterprise due to unpredictable pressures of the external environment, the probability of transition from state S_2 to S_3 is equal to:

$$P(3) = (0.129, 0.535, 0.18, 0.156) \times \begin{bmatrix} 0.2 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.1 & 0.2 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.6 & 0.1 \end{bmatrix} \quad (15)$$

$$= (0.1285, 0.54, 0.178, 0.1535)$$

The probability that being in the S_3 state of the production system due to the introduction of innovation during the transition to the S_4 state and the operation of the enterprise in a state of uncertainty is equal to:

$$P(4) = (0.1285, 0.54, 0.178, 0.1535) \times \begin{bmatrix} 0.2 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.1 & 0.2 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.6 & 0.1 \end{bmatrix} \quad (16)$$

$$= (0.1282, 0.54105, 0.17675, 0.154)$$

The general probability that determines the degree of readiness of the enterprise for innovation based on the data (Table 1 and Table 2) can be represented as a system of inequalities.

$$P(1) > P(2) > P(3) > P(4) \quad (17)$$

The presented system of inequalities is an indicator of the main diagnostic parameters of the readiness of enterprises for innovative activity, based on the obtained inequality, the priority, the main content and the level of changes are determined, a step-by-step plan is drawn up, an analysis of constraints and potential problems is carried out, strategies are developed and the determination of the necessary resources. With regard to determining the degree of readiness of enterprises to innovate in conditions of uncertainty of the influence of the

external environment, characterized by elements of randomness, the purpose of Markov chains is to search for such a combination of characteristics and parameters of innovation activity that would allow improving the mechanisms of diagnostics and decision-making in a visualized form at various levels. introducing innovations.

The orientation of the enterprise to the innovative path of development can be determined by the way of analyzing the capabilities and readiness of the enterprise for innovation. The use of the Markov chain reveals the essence and relationship of the main organizational parameters of the introduction of innovations to the likelihood of their manifestation in various options for the influence of the external environment. The readiness of an enterprise for innovation is determined not only by the existing potential, but also by a set of conditions that promote or hinder the implementation of innovations, which include factors of state regulation, international factors, strategic directions of innovation activities of enterprises, competitive advantage in the industry, disruption or stability of economic ties, market imbalance, political instability, etc.

Visualization of Markov chains in the finite-dimensional space of states of readiness for innovation (Table 1) is shown in Fig. 4.

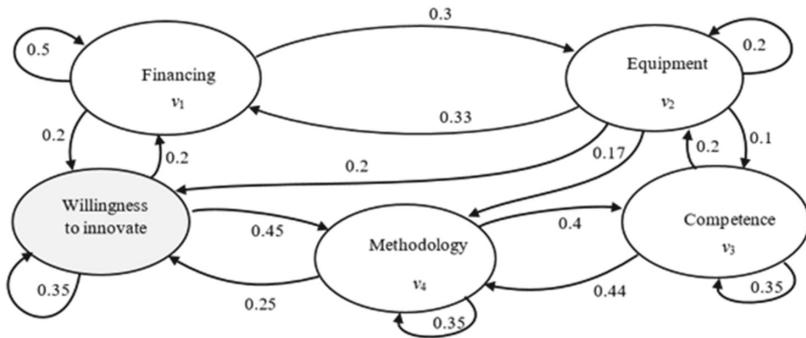


Fig. 4. Visualization of Markov chains in a finite-dimensional space of states of readiness for innovation

An integral assessment of the degree of readiness of enterprises for innovative activities is a convolution of individual indicators into a single generalized parameter that characterizes the directions of achieving maximum profit.

The advantage of the information management system for assessing the degree of readiness of enterprises for innovation using Markov chains is the ability to customize the system to any information situation. So in the example with four input variables v_i , the priority of the first and the sequence of subsequent steps can be changed. Behind the funding position v_1 there may be personnel competence v_3 which requires advanced training through courses, training seminars, trainings, etc. The acquisition of equipment v_2 is a time-saving operation

in the matrix of transition probabilities according to the ranking of priorities, can be located after the competence of personnel v_3 .

Other options for constructing Markov chains that implement information situations of the priority components of enterprises' readiness for innovation are shown in Fig. 5, Fig. 6, Fig. 7.

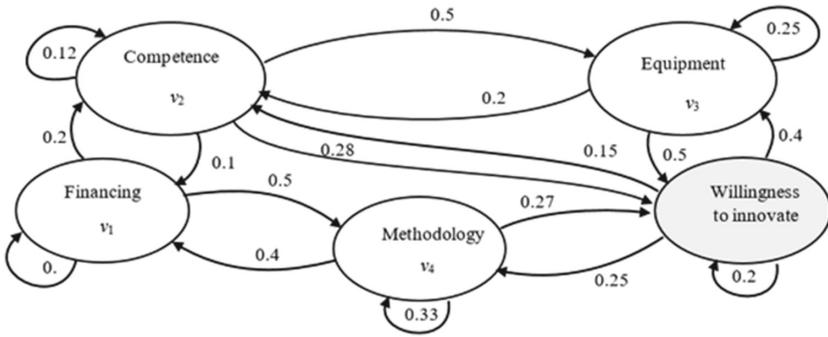


Fig. 5. Combinatorics of constructing Markov chains to assess the degree of readiness of enterprises for innovation with the main input variable - competence

Despite the complexity of Markov chains as a tool for determining the degree of readiness of enterprises for innovative activities, associated with the randomness and stochasticity of the process of innovative transformations, wandering and shifting the starting point of the report at all stages of implementation, choosing the number of input variables, hierarchy of sampling intervals, subjectivity of priorities, Markov chains are a powerful tool for studying the dynamic probabilities of processes with a quantitative implementation and practical orientation.

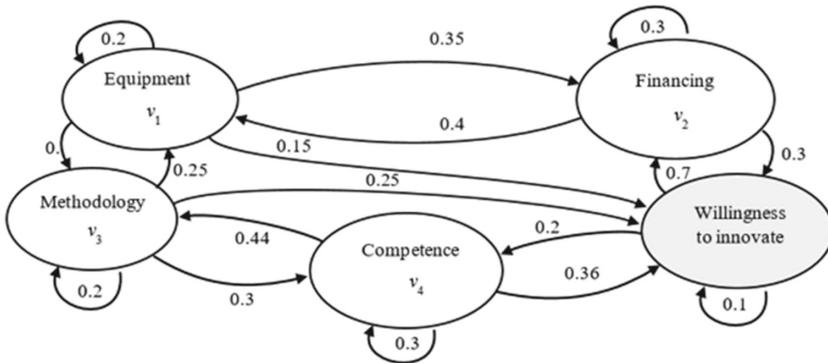


Fig. 6. Combinatorics of constructing Markov chains for assessing the degree of readiness of enterprises for innovation with the main input variable - equipment

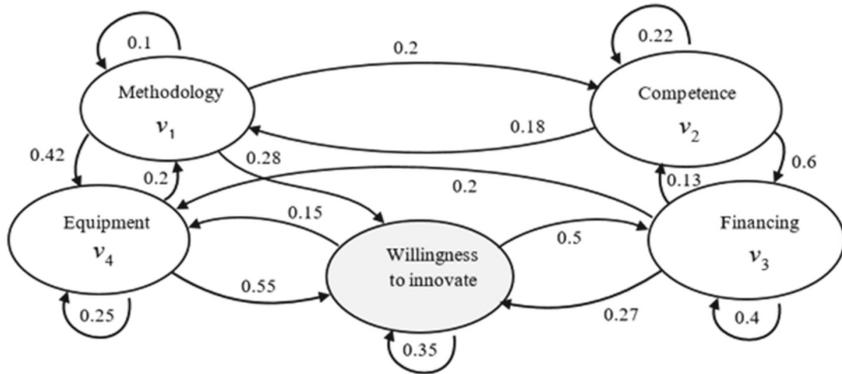


Fig. 7. Combinatorics of constructing Markov chains for assessing the degree of readiness of enterprises for innovation with the main input variable - methodology

To improve the accuracy of assessments of the degree of readiness of enterprises to innovate under conditions of uncertainty, a comprehensive analysis of various parameters and characteristics is required that determine different levels of innovation implementation and possible transformations of management mechanisms.

5 Conclusions

For a quantitative assessment of the degree of readiness of enterprises for innovation, it is necessary to use optimization and probabilistic methods using Markov chains. Markov chains are a powerful tool for probabilistic stochastic modeling of analytical data in a finite state space and an effective tool for assessing the readiness of enterprises for innovation, allowing us to see various options for solutions in a visualized form.

The features of Markov chain models allowing to take into account the multi-directional interests of the participants in innovative transformations, associated with the possibility of obtaining additional profit and with a constant share of the risk present in this process were presented.

The novelty of technologies for using Markov chains to assess the degree of readiness of enterprises for innovation is the hierarchy of sampling intervals and the replacement of the process time by a sequence of states of implementation of innovation, which allows to combine the results of forecasting at different levels of implementation of innovations with one general assessment of the degree of readiness of enterprises for innovation implementation of innovations.

With regard to determining the degree of readiness of enterprises to innovate under conditions of uncertainty of the influence of the external environment, characterized by elements of randomness, the of Markov chains using was provided to search for such a combination of characteristics and parameters of innovation activity that allow improving the mechanisms of diagnostics and

decision-making in a visualized form at various levels implementation of innovations. The use of Markov chains helps us to develop marketing strategies when introducing innovations.

References

1. Babichev, S., Skvor, J.: Technique of gene expression profiles extraction based on the complex use of clustering and classification methods. *Diagnostics* **10**(8) (2020). <https://doi.org/10.3390/diagnostics10080584>
2. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopolymers Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
3. Bilovodska, O., Kholostenko, A., Mandrychenko, Z., et al.: Innovation management of enterprises: Legal provision and analytical tools for evaluating business strategies. *J. Optim. Ind. Eng.* **14**, 71–78 (2021). <https://doi.org/10.22094/joie.2020.677820>
4. Chernev, V., Churyukin, V., Shmidt, A.: Modeling the economic sustainability of an enterprise using Markov chains with income. *Bull. South Ural State Univ.* **4**, 297–300 (2006)
5. Gagliardi, F., et al.: A probabilistic short-term water demand forecasting model based on the markov chain. *Water* **9**(7), 507 (2017)
6. Haque, S., Mengersen, K., Stern, S.: Assessing the accuracy of record linkages with Markov chain based Monte Carlo simulation approach. *J. Big Data* **8**(1), 1–25 (2021). <https://doi.org/10.1186/s40537-020-00394-7>
7. Huang, Q., et al.: A chan-vese model based on the Markov chain for unsupervised medical image segmentation. *Tsinghua Sci. Technol.* **26**(6), 833–844 (2021)
8. Khmaladze, E.: Testing hypothesis on transition distributions of a Markov sequence. *J. Stat. Plann. Inference* **215**, 72–84 (2021)
9. Kuznetsova, M.: Scientific and Innovative Activities, chap. Statistical Publication, p. 380. State Statistics Service of Ukraine (2020)
10. Kuznichenko, V., Lapshin, V.: Generalized scarcity exchange model for continuous processes with external control. *Econ. Manag.* **5**, 92–95 (2017)
11. Litvinenko, V.I., Burgher, J.A., Vyshemirskij, V.S., Sokolova, N.A.: Application of genetic algorithm for optimization gasoline fractions blending compounding. In: *Proceedings - 2002 IEEE International Conference on Artificial Intelligence Systems, ICAIS 2002*, pp. 391–394 (2002). <https://doi.org/10.1109/ICAIS.2002.1048134>
12. Ludwig, R., Pouymayou, B., Balermipas, P., et al.: A hidden markov model for lymphatic tumor progression in the head and neck. *Sci. Rep.* **11**(12261) (2021). <https://doi.org/10.1038/s41598-021-91544-1>
13. Lytvynenko, V., Lurie, I., Krejci, J., Voronenko, M., Savina, N., Taif, M.A.: Two step density-based object-inductive clustering algorithm. In: *CEUR Workshop Proceedings*, vol. 2386, pp. 117–135 (2019)
14. Obhiamo, J., Weke, P., Ngare, P.: Modeling Kenyan economic impact of corona virus in Kenya using discrete time Markov chains. *J. Financ. Econ.* **8**(2), 80–85 (2020)
15. Panarina, D.: Arrangement of markov breaking chains in the economy. *Vesnik Tyumen State Oil Gas Univ.* **11**(2(64)), 79–82 (2015)

16. Pysarenko, T., Kuranda, T., Kvasha, T., et al.: State of and Innovative Activity in Ukraine in 2020, chap. Statistical Publication, p. 40. State Statistics Service of Ukraine (2020)
17. Sharko, M., Gusarina, N., Petrushenko, N.: Information-entropy model of making management decisions in the economic development of the enterprises. *Adv. Intell. Syst. Comput.*, 304–314 (2019). <https://doi.org/10.1007/978-3-030-26474-1>
18. Sharko, M., Liubchuk, O., Fomishyna, V., et al.: Methodological support for the management of maintaining financial flows of external tourism in global risky conditions. *Commun. Comput. Inf. Sci.* (1158), 188–201 (2020). <https://doi.org/10.1007/978-3-030-61656-4>
19. Sharko, M., Lopushynskiy, I., Petrushenko, N., et al.: Management of tourists' enterprises adaptation strategies for identifying and predicting multidimensional non-stationary data flows in the face of uncertainties. *Advances in Intelligent Systems and Computing*, pp. 135–151 (2020). <https://doi.org/10.1007/978-3-030-54215-3>
20. Sharko, M., Shpak, N., Gonchar, O., et al.: Methodological basis of causal forecasting of the economic systems development management processes under the uncertainty. *Advances in Intelligent Systems and Computing* pp. 423–437 (2020). <https://doi.org/10.1007/978-3-030-54215-3>
21. Sharko, M., Doneva, N.: Methodological approaches to transforming assessments of the tourist attractiveness of regions into strategic managerial decisions. *Actual Problems of Economy* (8 (158)), 224–229 (2016)
22. Sharko, M., Sharko, A.: Innovative aspects of management of development of enterprises of regional tourism. *Actual Problems Econ.* **7**(181), 206–213 (2016)
23. Sherstennikov, Y.: Application of the Markov process model to the study of the economic efficiency of the firm. *Econ. Herald Donbass* **2**, 5–12 (2007)
24. Schmidt, A., Churyukin, V.: Markov models of economic systems. *Bull. South Ural State Univ.* **9**(3), 100–105 (2015)
25. Vorobyova, K.: The effect of brand perception in Malaysia's international airline industry during covid 19. *Ann. Soc. Sci. Manage. Stud.* **6**(4) (2021). <https://doi.org/10.19080/ASM.2021.06.555693>
26. Vorobyova, K.: The impact of individual work practices, social environment, managerial skills on workers' productivity: mediating role of international work experience. *Int. J. Pharmaceutical Res.* **13**, 26–27 (2021). <https://doi.org/10.31838/ijpr/2021.13.02.438>
27. Wang, L., Laird-Fick, H., Parker, C., et al.: Using Markov chain model to evaluate medical students' trajectory on progress tests and predict usmle step 1 scores a retrospective cohort study in one medical school. *BMC Med. Educ.* (21) (2021). <https://doi.org/10.1186/s12909-021-02633-8>
28. Zhao, Y., et al.: Spatio-temporal Markov chain model for very-short-term wind power forecasting. *J. Eng.* **2019**(18), 5018–5022 (2019)



Method to Find the Original Source of COVID-19 by Genome Sequence and Probability of Electron Capture

Yoshio Matsuki¹ , Aleksandr Gozhyj² , Irina Kalinina² ,
and Peter Bidyuk¹ 

¹ National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
matsuki@wdc.org.ua

² Petro Mohyla Black Sea National University, Nikolaev, Ukraine
alex.gozhyj@gmail.com

Abstract. The purpose of this study was to find the original source of envelope protein (spiked surface) of the Covid-19. It was assumed that the envelope protein was related to ordinary proteins like the human liver enzymes as possible original sources. A comparison was made on the genome sequences of the envelope protein and the human liver enzymes. The results of computational experiments showed that the longest sequence, common in both groups, was as follows: glutamine acid (e) - glutamine acid (e) - threonine (t) - glycine (g). Upon this finding further investigation was performed on the molecular structure of this sequence; and the probabilities of electron captures by the protons of the atoms were computed to determine which atoms could connect the amino acids using the approximation method taken from the quantum mechanics. The study was continued to identify which amino acid grew the genome sequence of the envelope protein differently from the human liver enzymes. And it was found that the electron capture by the proton of the atom could explain the process that formed the genome sequence of the Covid-19’s envelope protein out from the human liver enzymes. To our opinion this method could be used for identification of other candidate proteins so that to find the original source of the virus.

Keywords: Covid-19 · Envelope protein · Human liver enzymes · Amino acid · Genome sequence · Electron capture

1 Introduction

Currently, there is considerable interest in studies touching upon the underlying causes of Covid-19. One of the research areas is directed to the search for and modeling of the original genome sequence and testing of various hypotheses for the origin of this sequence. Various sources give different approaches to the study of this problem [3, 7, 9, 10, 13, 15, 16].

The study [2, 5] focuses on the problems of mathematical modeling of Covid-19. The paper provides for a comprehensive review of the challenges, problems, gaps and opportunities relevant to mathematical modeling of Covid-19 data and corresponding objectives. It gives a researcher wide view of problems, and modeling approaches, compares and discusses related methods and current progress in modeling Covid-19 epidemic transmission and dynamics, case identification and tracking, infection diagnosis and medical treatment. The methods of modeling considered involve mathematical and statistical approaches, epidemiological compartmental models, biomedical analysis as well as simulation and hybrid modeling. Availability of numerous approaches to modeling highlights the problem complexity and necessity for the search and hiring more effective models and treatment of the Covid-19 methods [1, 6, 8, 11, 12, 19].

The research paper [14, 18] considers the problem of predictive modeling of Covid-19 infection using statistical data from the Benelux union. Basically in this study the authors proposed the following two methodologies of modeling: a susceptible, exposed, infected, recovered, deceased (SEIRD) model that is based upon application of differential equations, and a long short-term memory (LSTM) deep learning neural network model. The Covid-19 epidemic caused the number of infected people to grow exponentially. The SEIRD compartmental model that included such variables as susceptible, exposed, infected (mild, severe, and critical), recovered, and deceased. The model parameters were computed with the data for the Benelux countries. The results achieved showed that SEIRD model predicted correctly several peaks for the three countries with higher RMSE for non-critical cases. The LSTM approach turned out to be capable to predict later peaks with low values of RMSE criterion. Some mismatch between real and simulated data was discovered that was explained by under-reporting of the number of cases, different initial conditions, as well as setting parameters. Generally, the model showed acceptable results and can be modified so that to take into consideration various phenomena like medical intervention, asymptomatic infection, mobility of people etc. The authors did not consider the problem of locating the original source of Covid-19.

The paper [4, 17] is based upon analysis of real time daily data and constructing the predictive model named as susceptible, exposed, infectious, recovered (SEIR). The modeling approach was used to forecast Covid-19 outbreak within and outside of China. The authors also collected and analyzed queried news to classify it into negative and positive sentiments. This approach helped to understand influence of the news on people behavior from political and economic point of view. The news queries during the study showed that there were more negative than positive articles (negative and positive sentiments). The positive sentiments were touching upon collaboration and strength of the individuals facing the epidemic, and the negative sentiments were related to the uncertainty and poor outcome like death. It was stressed that Covid-19 was still unclear disease what means that correct prediction can only be made after the end of the outbreak. The authors did not consider the problem of locating the original source of Covid-19.

Problem Statement. The research question was stated as follows: “What formed the genome sequence of the Covid-19’s envelope protein (spiked surface of the virus)?” At first, it was assumed that any ordinary protein such as human liver enzymes could be the original source of the envelope protein. Also comparisons were made on the genome sequences of the envelope protein and the human liver enzymes. As the result it was found that the longest match of the sequences was **e-e-t-g** (glutamine acid – glutamine acid – threonine – glycine), that are common in both the envelope protein and in the human liver enzymes.

It is proposed to investigate an approach in which the molecular structure of each of these four amino acids is examined in the longest **e-e-t-g** match. If a proton of an atom of the amino acid captured an electron of other amino acids atom, such phenomena could connect the amino acids to make longer genome sequence. With this assumption the probability of electron capture was calculated on selected atoms of each amino acid. An analysis was made to explain how the Covid-19’s envelope protein diverged from the human liver enzymes.

2 Material and Methods

2.1 Comparing Genome Sequences

Genome sequence data was taken from reference [2] for the Covid-19’s envelope protein (protein_id = “QIB84675.1”). The genome sequence is as follows:

mysfvseetgtlivnsvllflafvflvtlailtalrlcayccnivnsvlvpksfyvysrvknlnsrsvpdlly.

And from references [14, 17], the genome sequences for two types of human liver enzymes was taken, as shown in Table 1 and Table 2. The comparison was made on the genome sequence of the envelope protein with each of these human liver enzymes, by the Algorithm 1 shown below.

Table 1. Genome sequence of human liver enzymes-1 [14]

Enzyme No	Genome sequence
1	masstgdrsq avrhglrakv ltldgmnprv rrvveyavrgp ivqrroleq elrqgvkcpf
61	teviranigd aqamgqrpit flrqvlalcv npdllsspnf pddakraer ilqacgghsl
121	gaysvssgiq liredvaryi errdggipad pnnvflstga sdaivtvkl lvageghtrt
181	gvlpipqyp lysatlaelg avqvdyylde erawaldvae lhralgqard hcrpralcvi
241	npgnptgqvq trecieavir fafeerlfl adevyqdnvy aagsqhsfk kvlnemgppy
301	agqqelasfh stskgymgec gfrggyvevv nmdaavqqqm lkmsvrlcp pvpqgalldl
361	vvspaptdp sfaqfqaekq avlaelaaka klteqvfnea pgisnvpvqg amysfprvql
421	ppravaqaq lglapdmffc lrleetgic vvpqsgfgqr egtyhfrmti lppleklrll
481	leklsrfhak ftleys

Table 2. Genome sequence of human liver enzymes-2 [17]

Enzyme No	Genome sequence
1	mqraaalvrr gcgprtppsw grsqssaaae asavlkvrpe rsrreritl esmnpqvkvav
61	eyavrgpivl kageielelq rgikkpftvev iranigdaqa mgqqpitflr qvmalctypn
121	lldspsfped akkrarrilq acggnslgsy sasqgvncir edvaayitr dggvpadpdn
181	iylttgasdg istilkilvs gggksrtgvm ipipqyplys aviseldaiq vnyyldeenc
241	walnvnelrr avqeakdhcd pkvlciinp nptgqvqsrk ciedvihfaw eeklflade
301	vyqdnvyspd crfhsfkvvl yemgpeyssen velasfhsts kgymgecgvr ggymevinlh
361	peikgqlvkl lsvrlcppvs gqaamdivnv ppvageesfe qfsrekesvl gnlakkakt
421	edlfnqvpgi hcnplqgamy afrifipak aveaaqahqm apdmfycmkl leetgicvvp
481	gsgfgqregt yhfrmtilpp veklktvlqk vkdfhinfla kya

Algorithm 1: Comparison of genomic sequences

Input: Two groups of genome sequences: *genome11*, *genome22*
Output: Results of comparison of genomic sequences

```

l = 0;
for i = 1; i ≤ n do
    l = l + 1;
    Verification of the first genomic sequence: genome01;
    k = 0;
    p = 0;
    for j = 1; j ≤ m do
        k = k + 1;
        Verification of the second genomic sequence: genome02;
        if genome01 = genome02 then
            p = p + 1;
            Return matched genome sequence: genome01, matched
            number: p;
        end
    end
end
end

```

The above program shows an example, which looks for the matches of four consecutive amino acids. This program was used also to find three consecutive amino acids and two consecutive amino acids. In addition, an attempt was also made to find the matches of more than five consecutive sequences, however larger matched sequences were not detected.

2.2 Calculating the Probability of the Electron Capture

The algorithm for calculating the probability of electron capture was taken from [12]. The case was considered, in which a proton of an atom, for example Oxygen, captured an electron of another atom, for example Hydrogen, which

passed by the proton of the atom. Figure 1 shows the coordinates of two protons and an electron. Two protons are symmetrically located on both sides of the origin O of the coordinate. Here $\frac{1}{2}R$, and $(-\frac{1}{2}R)$, are the coordinates (geometric positions) of two protons that will capture the electron of another atom; R is the distance between two protons, and the positions of these protons are fixed. On the other hand, r is the position of electron in a plane polar coordinate system, and it changes as a function of geometric coordinate, x , where the relation holds: $-\frac{1}{2}R \leq x \leq \frac{1}{2}R$. We also assumed that the electron was initially attached to the proton at the coordinate of, $\frac{1}{2}R$; then the initial state of the electron had the form, $\varphi = (r - \frac{1}{2}R) \cdot \cos wx$, where $\cos wx$ was the Eigen-wave function, and w was the frequency of oscillation of the electron, representing its energy level.

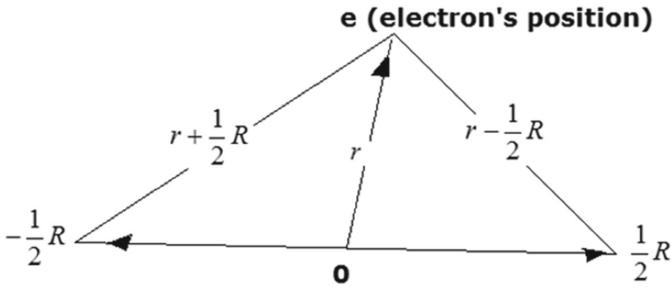


Fig. 1. The coordinates of the proton and the electron [12]

Also, in this research, three frequencies, w_1 , w_2 and w_3 , were set for simulating the lower, the middle and the higher energy levels, as shown in Fig. 2. Then, the probability of electron capture was calculated upon the Hamiltonian equation shown below:

$$H = T_e - C_1 \cdot \left(\frac{1}{r + \frac{1}{2}R}\right) \cdot \cos wx - C_2 \cdot \left(\frac{1}{r - \frac{1}{2}R}\right) \cdot \cos wx + C_3 \frac{1}{R} \cdot \cos wx, \quad (1)$$

where T_e is the kinetic energy of electron. Here it was assumed that the relative speed of proton was much slower than the electron's speed. Therefore, the geometry of an electron and protons was the main focus, not time dependency of the system; and T_e is a set as a unity (one).

When the electron is attached to one proton at the coordinate of $(+\frac{1}{2}R)$ as its initial state, the wave function is, $(r - \frac{1}{2}R) \cdot \cos wx$; but it changes to $(r + \frac{1}{2}R) \cdot \cos wx$ when the electron is transferred to another proton located at $(-\frac{1}{2}R)$, as the process of the charge exchange.

When the reference [12] was published in 1969, personal computers were not available, therefore the reference [12] further described the algorithm in mathematical forms with calculus, and predicted that the squared module of the

coefficient C_1 for $\left(-\frac{1}{r + \frac{1}{2}R} \cdot \cos wx\right)$ gave the probability of charge exchange (the electron capture). However, in this research a personal computer was used to calculate the coefficients, C_1 , C_2 , and C_3 with the following algorithm of matrix algebra:

$$H = T_e - X \cdot c, \tag{2}$$

where, X consists of three column vectors:

$$\left(-\frac{1}{r + \frac{1}{2}R} \cdot \cos wx\right), \left(-\frac{1}{r - \frac{1}{2}R} \cdot \cos wx\right), \text{ and } \left(+\frac{1}{R} \cdot \cos wx\right);$$

and vector is as follows:

$$c = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}. \tag{3}$$

Then a constraint is a set:

$$X'H = 0, \tag{4}$$

so that:

$$X'(T_e - Xc) = 0, \tag{5}$$

where X' is the transpose matrix of X .

Thus, we have:

$$X'Xc = X'T_e, \tag{6}$$

$$c = (X'X)^{-1}X'T_e, \tag{7}$$

and

$$V(c) = \hat{\sigma}^2(X'X)^{-1}, \tag{8}$$

where $V(c)$ is the variance of c , where:

$$\hat{\sigma}^2 = e'e/(n - 1), \tag{9}$$

$$e = M \cdot T_e, \tag{10}$$

$$M = I - X(X'X)^{-1}X', \tag{11}$$

where n is the number of rows of each column of X (in this simulation $n = 25$); l is the number of columns of X (in this simulation, $l = 3$); I is a 25×25 unit

matrix; $(X'X)^{-1}$ is the inverse matrix of $X'X$, and e' is the transpose vector of e . Finally, c is the coefficient vector, and $\sigma = \sqrt{V(c)}$ is the standard error of the coefficient.

For this simulation the values of R were assigned as the diameters of Hydrogen, Carbon, Nitrogen and Oxygen, as shown in the Table 3, and $r = \sqrt{\left(\frac{R}{2}\right)^2 + x^2}$, where x was the distance from origin O toward $(-\frac{1}{2}R)$ and toward $\frac{1}{2}R$ in Fig. 1, where the origin O was located at $x = 13$; and, $(-\frac{1}{2}R)$ was at $x = 1$, and $\frac{1}{2}R$ was at $x = 25$.

Note: According to [12], p. 84, “Capture of an Electron by a Proton (Charge Exchange)” of the Chap. 2.4 “Adiabatic Perturbations”, R is the distance between the two protons of Fig. 1. Here an assumption was made as if two same atoms, which were centered by each of two protons, were located next to each other: therefore, $R = 2 \times re$, where re is empirically measured radius of Table 3.

Table 3. Input data of R for the simulation

Chemical element	Empirically measured radius (re) in pico-meters [12]	R
Hydrogen atom	25	50
Carbon atom	70	140
Nitrogen atom	65	130
Oxygen atom	60	120

Then the input data were made on R , r , $\cos wx$ and $\sin wx$ as shown in Fig. 2. The case of $\sin wx$ was also calculated during this research, but it was eliminated from this report due to the less significance of the calculated standard error of the coefficient.

3 Results

Table 4 shows the matched genome sequences of the envelope protein and of each of two types of the liver enzymes.

The genome sequence of **e-e-t-g** (glutamine acid – glutamine acid – threonine – glycine) was the longest matched sequence found both in the envelope protein of Covid-19 and human liver enzymes. Upon each of the molecular structures of **e** (glutamine acid), **g** (glycine) and **t** (threonine), the probability of electron capture (charge exchange) by the proton of each of Hydrogen, Carbon, Nitrogen and Oxygen atoms was calculated. Figure 3 shows the molecular structures of **e** (glutamine acid), **t** (threonine) and **g** (glycine).

Table 4. Matched genome sequences in the Covid-19's envelope protein and human liver enzymes

With Liver enzyme-1				With Liver enzyme-2			
Sequence	Number	Sequence	Number	Sequence	Number	Sequence	Number
eetg	1	vv	3	eetg	1	vp	3
eet	1	vs	4	eet	1	il	4
etg	1	Li	2	etg	1	lf	2
ysf	1	vn	4	ilt	1	lr	2
vfl	1	sv	2	psf	1	ni	2
llv	2	Vf	2	lfl	1	ns	2
lfl	1	lv	6	fl	1	ps	2
fl	1	Tl	2	vys	1	sv	2
rlc	1	ll	5	svl	1	vv	2
pdll	2	Lt	2	rlc	1	vy	2
mys	1	vk	4	lvk	1	af	1
lrl	2	Dl	2	ll	12	ai	1
psf	1	Vl	2	sr	8	ay	1
tla	1	vy	2	sf	8	cn	1
dll	1	vp	2	ys	8	dl	1
ll	21	sr	2	la	6	et	1
fl	8	vt	1	lv	11	fy	1
la	14	ai	1	vk	6	gt	1
al	6	ay	1	vn	8	iv	2
ys	8	ni	1	tg	4	kp	1
rv	6	nv	2	vl	6	ln	1
tg	5	sl	1	vs	4	my	1
lr	5	kp	1	al	3	rl	1
iv	4	lf	1	ee	3	se	1
Vl	4	af	1	fl	6	sl	1
Sf	8	yv	1	lc	3	tl	2
Rl	4	my	1	lt	3	yc	1
ee	3	et	1	nl	3	-	-
Lc	3	gt	1	nv	3	-	-
ss	3	cn	1	pd	3	-	-
pd	3	ps	1	ss	3	-	-

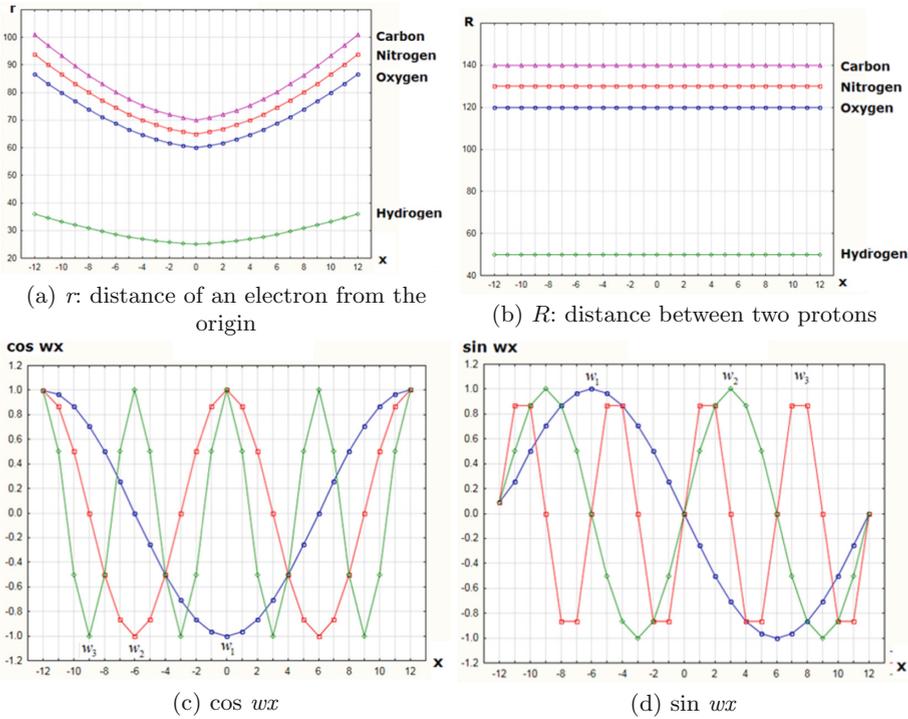


Fig. 2. (a, b, c, d) Input data for the simulation: r , R , $\cos wx$, and $\sin wx$

Figure 4 shows the calculated probability of electron captures with $\cos wx$ for each of the protons of Hydrogen, Carbon, Nitrogen and Oxygen atoms, and Tables 5, 6, 7 and 8 shows the calculated numeric values of the coefficients and the standard errors of the coefficients as well as the squared modules of the coefficients and the probabilities of the electron captures by the atoms. The coefficients and the standard errors of the coefficients with $\sin wx$ were also calculated within this research, but their calculated results were omitted from this report. Because the calculated standard errors of the coefficients of $\left(-\frac{1}{r - \frac{1}{2}R} \sin wx\right)$, were larger than the calculated values of the coefficients, which meant that these calculated coefficients were not significant. On the other hand, the calculated standard errors of the coefficients of $\left(-\frac{1}{r + \frac{1}{2}R} \cos wx\right)$, were smaller than the values of the coefficients. Therefore, the case of $\cos wx$ was more significant than the case of $\sin wx$. The difference between the calculated results with $\sin wx$ and $\cos wx$ was resulted from the geometrical distributions of $\sin wx$ and $\cos wx$ (shown in Fig. 2) within the geometrical framework of the model shown in Fig. 1. Now the probability of electron capture by an atom's proton was calculated by the squared module of the coefficient of $\left(-\frac{1}{r + \frac{1}{2}R} \cos wx\right)$

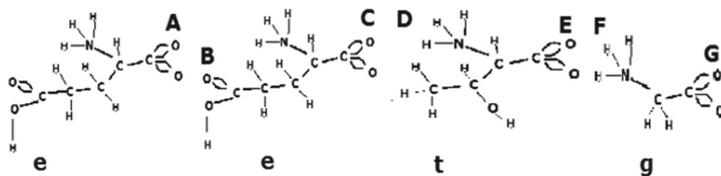


Fig. 3. Molecular structures of **e** (glutamine acid), **t** (threonine) and **g** (glycine)

divided by total of the squared modules of the coefficients of $\left(-\frac{1}{r + \frac{1}{2}R} \cos wx\right)$, $\left(-\frac{1}{r - \frac{1}{2}R} \cos wx\right)$, and $\left(+\frac{1}{R} \cos wx\right)$.

Table 5. Calculated probability of electron captures with $\cos wx$ for Hydrogen

Chemical element			$-\frac{1}{r + \frac{1}{2}R} \cdot \cos wx$	$-\frac{1}{r - \frac{1}{2}R} \cdot \cos wx$	$+\frac{1}{R} \cdot \cos wx$
Hydrogen	Coefficient C_i . Inside of the bracket: standard error of the coefficient	w_1	780,4 (252,2)	$1,622 \cdot 10^{-4}$ ($5,781 \cdot 10^{-4}$)	714,2 (229,6)
		w_2	335,9 (239,2)	$8,066 \cdot 10^{-4}$ ($5,647 \cdot 10^{-4}$)	305,0 (217,6)
		w_3	258,2 (252,2)	$7,345 \cdot 10^{-4}$ ($5,781 \cdot 10^{-4}$)	234,6 (229,6)
	$ C_i ^2$	w_1	$6,091 \cdot 10^5$	$2,632 \cdot 10^{-8}$	$5,101 \cdot 10^5$
		w_2	$1,129 \cdot 10^5$	$6,506 \cdot 10^{-7}$	$9,305 \cdot 10^4$
		w_3	$6,666 \cdot 10^4$	$5,395 \cdot 10^{-7}$	$5,506 \cdot 10^4$
	Probability $\frac{ C_i ^2}{ C_1 ^2 + C_2 ^2 + C_3 ^2}$	w_1	0,5442	$2,351 \cdot 10^{-14}$	0,4558
		w_2	0,5484	$3,160 \cdot 10^{-12}$	0,4520
		w_3	0,5477	$4,432 \cdot 10^{-12}$	0,4523

Note: $i = 1$ for $\left(-\frac{1}{r + \frac{1}{2}R} \cdot \cos wx\right)$, and $i = 2$ for $\left(-\frac{1}{r - \frac{1}{2}R} \cdot \cos wx\right)$, and $i = 3$ for $\left(+\frac{1}{R} \cdot \cos wx\right)$.

Upon the calculated probabilities of electron captures, two possibilities were examined on the **e-e-t-g** connection as shown in Fig. 5 (Case-1) and Fig. 6 (Case-2). (Note: In Fig. 5 and Fig. 6, the direction of **t** (threonine), between **D** and **E**, is opposite from each other. The reason is explained below.)

Table 6. Calculated probability of electron captures with $\cos wx$ for Carbon

Chemical element			$-\frac{1}{r + \frac{1}{2}R} \cdot \cos wx$	$-\frac{1}{r - \frac{1}{2}R} \cdot \cos wx$	$+\frac{1}{R} \cdot \cos wx$
Carbon	Coefficient C_i . Inside of the bracket: standard error of the coefficient	w_1	$2,185 \cdot 10^3$ (376,1)	$1,637 \cdot 10^{-4}$ ($3,650 \cdot 10^{-4}$)	$2,000 \cdot 10^3$ (341,2)
		w_2	942,2 (669,7)	$8,149 \cdot 10^{-4}$ ($5,692 \cdot 10^{-4}$)	855,4 (609,4)
		w_3	723,6 (706,2)	$7,408 \cdot 10^{-4}$ ($5,827 \cdot 10^{-4}$)	657,6 (643,1)
	$ C_i ^2$	w_1	$4,776 \cdot 10^6$	$2,678 \cdot 10^{-8}$	$4,000 \cdot 10^6$
		w_2	$8,878 \cdot 10^5$	$6,640 \cdot 10^{-7}$	$7,318 \cdot 10^5$
		w_3	$5,236 \cdot 10^5$	$5,488 \cdot 10^{-7}$	$4,324 \cdot 10^5$
	Probability $\frac{ C_i ^2}{ C_1 ^2 + C_2 ^2 + C_3 ^2}$	w_1	0,5442	$3,052 \cdot 10^{-15}$	0,4558
		w_2	0,5482	$4,100 \cdot 10^{-13}$	0,4518
		w_3	0,5477	$5,741 \cdot 10^{-13}$	0,4523

Table 7. Calculated probability of electron captures with $\cos wx$ for Nitrogen

Chemical element			$-\frac{1}{r + \frac{1}{2}R} \cdot \cos wx$	$-\frac{1}{r - \frac{1}{2}R} \cdot \cos wx$	$+\frac{1}{R} \cdot \cos wx$
Nitrogen	Coefficient C_i . Inside of the bracket: standard error of the coefficient.	w_1	$2,0293 \cdot 10^3$ (349,2)	$1,637 \cdot 10^{-4}$ ($3,650 \cdot 10^{-4}$)	$1,8570 \cdot 10^3$ (316,8)
		w_2	874,9 (621,9)	$8,148 \cdot 10^{-4}$ ($5,692 \cdot 10^{-4}$)	794,3 (565,9)
		w_3	671,9 (655,7)	$7,408 \cdot 10^{-4}$ ($5,827 \cdot 10^{-4}$)	610,6 (597,2)
	$ C_i ^2$	w_1	$4,118 \cdot 10^6$	$2,678 \cdot 10^{-8}$	$3,449 \cdot 10^6$
		w_2	$7,654 \cdot 10^5$	$6,639 \cdot 10^{-7}$	$6,309 \cdot 10^5$
		w_3	$4,514 \cdot 10^5$	$5,487 \cdot 10^{-7}$	$3,728 \cdot 10^5$
	Probability $\frac{ C_i ^2}{ C_1 ^2 + C_2 ^2 + C_3 ^2}$	w_1	0,5442	$3,539 \cdot 10^{-15}$	0,4558
		w_2	0,5482	$4,755 \cdot 10^{-13}$	0,4518
		w_3	0,5477	$6,658 \cdot 10^{-13}$	0,4523

Table 8. Calculated probability of electron captures with $\cos wx$ for Oxygen

Chemical element			$-\frac{1}{r + \frac{1}{2}R} \cdot \cos wx$	$-\frac{1}{r - \frac{1}{2}R} \cdot \cos wx$	$+\frac{1}{R} \cdot \cos wx$
Oxygen	Coefficient C_i Inside of the bracket: standard error of the coefficient.	w_1	$1,873 \cdot 10^3$ (322,3)	$1,624 \cdot 10^{-4}$ (3,622 · 10 ⁻⁴)	$1,714 \cdot 10^3$ (292,4)
		w_2	807,5 (574,0)	$8,085 \cdot 10^{-4}$ (5,649 · 10 ⁻⁴)	733,1 (522,3)
		w_3	620,2 (620,2)	$7,351 \cdot 10^{-4}$ (7,351 · 10 ⁴)	563,6 (551,2)
	$ C_i ^2$	w_1	$3,509 \cdot 10^6$	$2,6374 \cdot 10^{-8}$	$2,938 \cdot 10^6$
		w_2	$6,520 \cdot 10^5$	$6,537 \cdot 10^{-7}$	$5,375 \cdot 10^5$
		w_3	$3,846 \cdot 10^5$	$5,404 \cdot 10^{-7}$	$3,176 \cdot 10^5$
	Probability $\frac{ C_i ^2}{ C_1 ^2 + C_2 ^2 + C_3 ^2}$	w_1	0,5442	$4,091 \cdot 10^{-15}$	0,4559
		w_2	0,5482	$5,496 \cdot 10^{-13}$	0,4518
		w_3	0,5477	$7,696 \cdot 10^{-13}$	0,4523

The logics to formulate these connections are as follows.

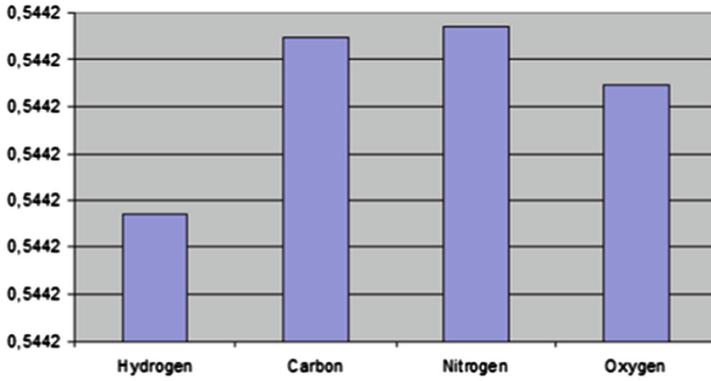
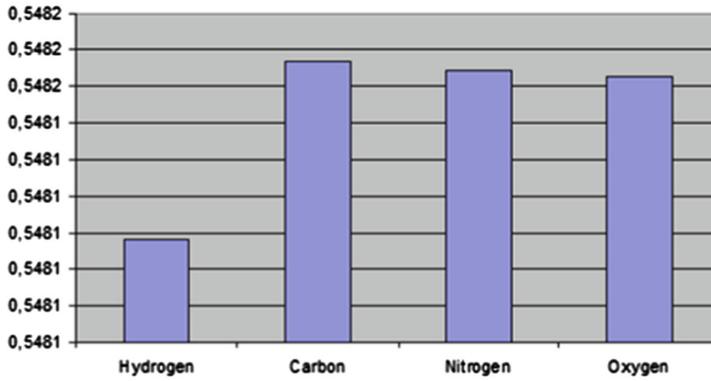
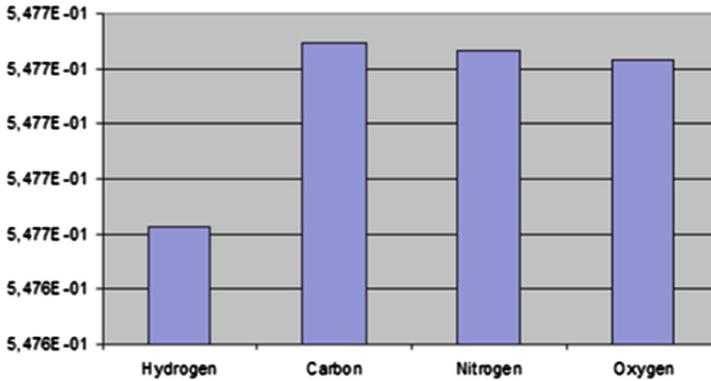
1. *Connection from e to e.* The both sides of **e** (glutamine acid) are edged by Oxygen atoms. Therefore, the **e-e** connection occurs between two Oxygen atoms (**A** and **B**), where one electron of one Oxygen atom of one glutamine acid (**A**) is to be captured by another Oxygen atom of another glutamine acid (**B**). In Fig. 4 and Tables 5, 6, 7 and 8 the calculated probabilities of electron capture by Oxygen atom is higher with the frequency of the electron wave of w_2 than, w_1 . In other words, the probability of electron capture by Oxygen atoms is higher if the energy level of the to-be-captured electron is higher in a certain range. In this case the captured electron belongs to the outer shell of the larger atom that is Oxygen, not of smaller atom.
2. *Connection from e to t.* The **e-t** connection must occur at Oxygen atoms of **e** to either of Oxygen atom or Carbon atom of **t** (threonine), by dropping Hydrogen atom from the Carbon atom, because Hydrogen atom cannot be connected with two atoms. In Fig. 5 and Fig. 6, the dropped Hydrogen atom from the Carbon atom is marked in a circle. This Hydrogen atom may be combined with the other free atom such as Oxygen to form OH. Figure 4 and Tables 5, 6, 7 and 8 show that the probability of the electron capture by the proton of Hydrogen atom is lower; and, Table 9 shows that the **e-t** connection was observed only once in both of Covid-19's envelope and the human liver enzymes. This observation suggests that the connection from **e** to **t** is a rare occurrence; therefore, Case-2 (Fig. 6) is a reasonable possibility. On the other hand, Fig. 5 shows that the **e-t** connection occurs between an Oxygen atom and another Oxygen atom, which suggests the higher probability of electron capture; and, it conflicts with the rare observation in Table 9.

Note: **e**: Glutamine Acid; **t**: Threonine; **g**: Glycine; **l**: leucine; **c**: cysteine; **i**: isoleucine.

3. *Connection from t to g.* Table 9 shows that the **t-g** connection was observed only once in the Covid-19's envelope alone, while 5 times in the liver enzyme-1, and 4 times in the liver enzyme-2; therefore, it is not a rare occurrence in the human liver enzymes, and the Oxygen-Oxygen connection is reasonably explained by the high probability of electron capture.
4. *Connection from e-e-t-g to the other amino acids in the Covid-19's envelope protein.* In the genome sequence of Covid-19's envelope protein alone [2], the sequence **e-e-t-g** continued as **e-e-t-g-t-l** ... However, **g** (glycine) had no possibility of further connection to **t** (threonine) unless the Hydrogen of **g** (glycine), located on the right-hand side of the **e-e-t-g** chain in Fig. 6, was dropped out from the Nitrogen atom so that the Nitrogen atom could be directly connected to the other atom. Table 9 shows that the **g-t** connection is observed only once in both of the envelope protein and the liver protein. However, the sequence **e-e-t-g-t** expands to **e-e-t-g-t-l** in the Covid-19's envelope protein because **l** (leucine) includes Oxygen atom (see Fig. 7 below) that holds the high probability of electron capture.
5. *Connection from e-e-t-g to the other amino acids in the human liver enzymes.* In the human liver enzyme-1 (Table 2) and the enzyme-2 (Table 3), the sequence of **e-e-t-g** is followed by **i-c** ... to form **e-e-t-g-i-c** ..., which doesn't appear in the Covid-19's envelope protein [2]. Table 9 shows that the **g-i** connection appears 4 times in both of the liver enzyme-1 and the liver enzyme-2. Figure 7 shows that **i** (isoleucine) holds Oxygen atom, and it can further connect **e-e-t-g-i** to **c** (cysteine), which also holds Oxygen atoms. Therefore, it is reasonable to observe **e-e-t-g-i-c** ... in the ordinary human liver enzymes.
6. *Amino acid that diverges the envelope protein from the human liver enzymes.* In the envelope protein, **g** (glycine) is the last amino acid of the chain **e-e-t-g**, and it holds Hydrogen atom. The Hydrogen atom has the lower probability of electron capture, but **t** (threonine) holds Oxygen atom that has the higher probability of electron capture; therefore, if the Hydrogen atom is dropped

Table 9. Two consecutive amino acids

Amino acid	Connected amino acid on the right-hand side	Number of observations		
		Covid-19's envelope alone	Liver enzyme-1 alone	Liver enzyme-2 alone
e	e-e	1	3	4
e	e-t	1	1	1
t	t-g	1	5	4
g	g-t	1	1	1
t	t-l	2	3	1
g	g-i	0	4	4
i	i-c	0	1	1

(a) w_1 (b) w_2 (c) w_2 Fig. 4. Probabilities of electron captures by the atoms with $\cos wx$

out from the Nitrogen atom, **t** (threonine) can be further connected to **l** (leucine) that also holds Oxygen atoms. On the other hand, in the human liver enzymes, **g** (glycine) of the chain **e-e-t-g** connects to **i** (isoleucine) because it holds Oxygen atom, and it further connects to **c** (cysteine) which also holds Oxygen atoms. In fact, Table 9 shows that the connection of **g-i** is not observed in the envelope protein, but it is observed 4 times in the liver enzymes.

The result of the analysis shows that the probability of electron capture by proton explains the difference between the genome sequences of the Covid-19's envelope protein and the human liver enzymes.

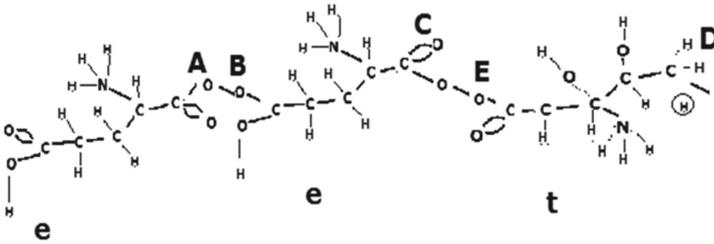


Fig. 5. Possible connections of e-e-t-g: Case-1

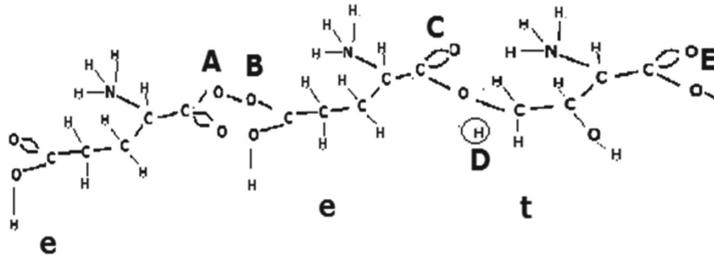


Fig. 6. Possible connections of e-e-t-g: Case-2

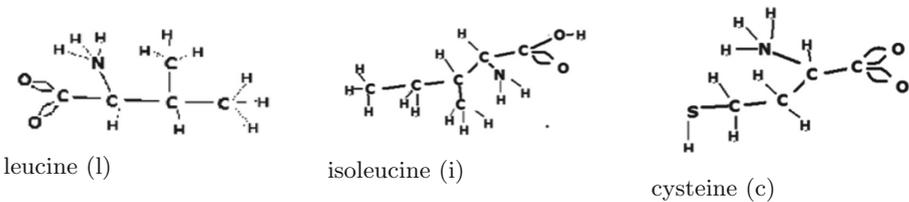


Fig. 7. Molecular structures of leucine (l), cysteine (c) and isoleucine (i)

4 Conclusions

The result of this analysis shows that the formation of Covid-19's envelope protein is explained by the probability of electron capture by the protons of atoms. For example, the sequence **e-e-t-g** is common in both of the envelope protein and the human liver enzymes; but, after this sequence, the envelope protein expands its sequence to **e-e-t-g-t-l- ...**, while the liver enzymes continue their sequences differently to **e-e-t-g-i-c. ...** This difference could be explained by the probabilities of the electron capture by the atoms in the structures of the amino acids.

Fig. 4 shows that the atoms increase their capabilities of electron capture if the energy level of the to-be-captured electron in the other atom is within a certain level. In order to explain such energy level, there are two possibilities. First, if the electron is in the outer shell of relatively larger atoms such as Oxygen or Carbon in comparison with the smaller atom such as Hydrogen, the probability of electron capture by the other atoms becomes higher. The other possibility is that the electron's energy state is excited by some reasons such as the scattering by photons; but this second possibility should be investigated further in the future research. This research demonstrated a possible procedure to find the origin of Covid-19 formation by following the steps:

1. the comparison of genome sequence of Covid-19 with ordinary proteins, as a candidate of the origin, to find the longest matched genome sequence;
2. the investigation of the molecular structure of each amino acid to check if the logic of the probability of electron capture works (for example, whether or not, the smaller probability of electron capture produces less number of the connections in the genome sequence of Covid-19 and in the candidate protein for the possible origin);
3. the assessment of the logic on how the electron capture creates the chains of the amino acids of the virus and of the candidate protein.

As the result of this research, it was explained that the Covid-19's envelope protein could emerge from the human liver enzymes. However, the further investigation is needed upon studying different candidate proteins.

References

1. Aleta, A., Martin-Corral, D., Piontti, A., Ajelli, M., Litvinova, M., Chinazzi, M.: Modelling the impact of testing, contact tracing and household quarantine on second waves of covid-19. *Nat. Hum. Behav.* **4**, 964–971 (2020). <https://doi.org/10.1038/s41562-020-0931-9>
2. Bastola, A., et al.: The first 2019 novel coronavirus case in Nepal. *Lancet. Infect. Dis.* **3**, 279–280 (2020). [https://doi.org/10.1016/S1473-3099\(20\)30067-0](https://doi.org/10.1016/S1473-3099(20)30067-0)
3. Bidyuk, P., Gozhyj, A., Matsuki, Y., Kuznetsova, N., Kalinina, I.: Modeling and forecasting economic and financial processes using combined adaptive models. In: *Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2020*, vol. 1246, pp. 395–408 (2021). https://doi.org/10.1007/978-3-030-54215-3_25

4. Binti-Hamzah, F., Lau, C., Nazri, H., Ligot, D., Lee, G., Tan, C.: Corona tracker: World-wide covid-19 outbreak data analysis and prediction. World Health Organization (2020). <https://doi.org/10.2471/BLT.20.255695>
5. Cao, L., Liu, Q.: Covid-19 modeling: A review. COVID-19 Modeling, Univ. Technol. Sydney **1**(1), 1–73 (2021). <https://doi.org/10.1145/374308.374363>
6. Chen, Y., Lu, P., Chang, C., Liu, T.: A time-dependent sir model for covid-19 with undetectable infected persons. IEEE Trans. Network Sci. Eng., 3279–3294 (2020). <https://doi.org/10.1109/TNSE.2020.3024723>
7. Clifford, S., Pearson, C., Klepac, P., Van Zandvoort, K., Quilty, B.: Effectiveness of interventions targeting air travellers for delaying local outbreaks of sars-cov-2. Travel Med **27**(5), taaa068 (2020). <https://doi.org/10.1093/jtm/taaa068>
8. Desgraupes, B.: Coronavirus disease 2019. (covid-19) pandemic: Increased transmission in the EU/EEA and the UK (2020). <https://www.ecdc.europa.eu/sites/default/files/documents/RRR-seventh-updateOutbreak-of-coronavirus-disease-COVID-19.pdf>
9. Giordano, G., Blanchini, F., Bruno, R., Colaneri, P., Di Filippo, A., Di Matteo, A.: Modelling the covid-19 epidemic and implementation of population-wide interventions in Italy. Nat. Med. **26**, 855–860 (2020). <https://doi.org/10.1038/s41591-020-0883-7>
10. Gozhyj, A., Kalinina, I., Gozhyj, V., Danilov, V.: Approach for modeling search web-services based on color petri nets. In: DSMP 2020. Communications in Computer and Information Science, vol. 1158, pp. 525–538 (2020). <https://doi.org/10.1007/978-3-030-61656-4-35>
11. Hinch, R., Probert, W., Nurtay, A., Kendall, M., Wymant, C., Hall, M.: Openabm-covid19 - an agent-based model for non-pharmaceutical interventions against covid-19 including contact tracing. PLoS Comput. Biol. **17**(7), e1009146 (2021). <https://doi.org/10.1371/journal.pcbi.1009146>
12. Migdal, A., Krainov, V.: Frontiers in physics. In: Approximation Methods in Quantum Mechanics, p. 146. W.A. Benjamin Inc., New York (1969)
13. Piccolomini, E., Zama, F.: Monitoring Italian covid-19 spread by an adaptive seird model. medRxiv, pp. 1–15 (2020). <https://doi.org/10.1101/2020.04.03.20049734>
14. Sah, R., Rodriguez-Morales, A., Jha, R., Chu, D., Gu, H., Peiris, M., Bastola, A., Lal, B., Ojha, H., Rabaan, A.: Complete genome sequence of a 2019 novel coronavirus (sars-cov-2) strain isolated in nepal. Microbiol Resour Announc. **9**(11), e00169-20 (2020). <https://doi.org/10.1128/MRA.00169-20>
15. Shen, M., Peng, Z., Guo, Y., Xiao, Y., Zhang, L.: Lockdown may partially halt the spread of 2019 novel coronavirus in Hubei province, China. Infect. Dis. **96**, 503–505 (2020). <https://doi.org/10.1016/j.ijid.2020.05.019>
16. Slater, J.: Atomic radii in crystals. J. Chem. Phys. **41**(10) (1964). <https://doi.org/10.1063/1.1725697>
17. Sohocki, M., et al.: Human glutamate pyruvate transaminase (GPT): localization to 8q24.3, cdna and genomic sequences, and polymorphic sites. J. Genomics **40**(2), 247–252 (1997). <https://doi.org/10.1006/geno.1996.4604>
18. Sustersic, T., et al.: Epidemiological predictive modeling of covid-19 infection: development, testing, and implementation on the population of the benelux union. Public Health **9**(22), 727274 (2021). <https://doi.org/10.3389/fpubh.2021.727274>
19. Yang, R., Blaileanu, G., Hansen, B., Shuldiner, A., Gong, D.: CDNA cloning, genomic structure, chromosomal mapping, and functional expression of a novel human alanine aminotransferase. J. Genomics **79**(3), 445–450 (2002). <https://doi.org/10.1006/geno.2002.6722>



Leader-Follower Strategy of Fixed-Wing Unmanned Aerial Vehicles via Split Rejoin Maneuvers

Roneel Chand¹✉, Krishna Raghuwaiya², and Jito Vanualailai²

¹ Fiji National University, Suva, Fiji

roneel.chand@fnu.ac.fj

² The University of the South Pacific, Suva, Fiji

{krishna.raghuwaiya,jito.vanualailai}@usp.ac.fj

Abstract. In this paper, we adopt the Lyapunov- based Control Scheme (LbCS) architecture to propose a motion planner for several fixed-wing Unmanned Aerial Vehicles (UAVs). We implement the leader-follower formation type here, where the flock has to navigate in a workspace cluttered with obstacles with respect to its leader to reach its pre-defined targets. In our case, the obstacles will be spherical fixed obstacles and the moving aircrafts in the swarm itself becomes the obstacles for all the other members. This needs to be avoided to successfully achieve the task. The flock navigates the environment in its pre-defined formation and moves towards its target. In the event of an obstacle, the flock splits and rejoins later in a safer location, regaining the desired shape. A set of nonlinear acceleration-based controllers using the Lyapunov-based Control Scheme are designed to achieve this task successfully. The controllers presented will guarantee the UAVs coordinate their motion in a well-planned manner and make sure the aircraft converge to their desired target while avoiding obstacles intersecting their path. The computer-generated simulations of a number of virtual scenarios have been looked at where different predefined formations of the flock have been designed. These simulations show the effectiveness of the proposed nonlinear acceleration control laws, revealing the simplicity and effectiveness of the control technique used. The paper finally ends with a conclusion and future work recommendations in this area.

Keywords: Lyapunov · Fixed-wing UAVs · Flock · Split-rejoin · Leader-follower

1 Introduction

In today's world, robotic systems are considered one of the most powerful and important tools to carry out tasks that are not safe or impossible to be done by humans. Recent research has shown that groups of robots are used for tasks which cannot be carried out by single robots [2, 19]. In fact, this turned out to be more effective when compared to single robots [3, 4, 9]. Working with multiple

robots (Co-operative Intelligent Transport System (C-ITS)) is usually carried out in a specific way or formation where robots need to coordinate its motion in a cooperative manner. The added advantages of C-ITS are that it is cheaper when compared to single vehicles, it covers a larger area in a shorter time and it also improves task execution [13, 16].

Recent research has placed a lot of emphasis on formation control of multi-agent systems. Sharma in [15] once said that formation control is actually controlling the position and orientation of groups of agents while maintaining their relative locations. Formation control by design in multi-agent systems is adopted from many real-life situations, and Nature is highly praised for this. Behaviours such as schools of fish, herds of land animals, and flocks of birds, to name a few, are highly used in multi-agent systems for formation control. This outstanding behavior not only achieved great success in the multi-agent system but also brought more safety and strength [1, 7, 20]. Multi-agent systems work in a specific manner where the flock splits in case of obstacles or danger and rejoins at a safer distance. Multi-agent systems also share loads and execute tasks more and perform more safely when compared to single or split robots [14, 18].

Usually we look at the formation shape, formation type, and robotic control when we look at the formation control of multi-agent systems in general. Recent researchers have greatly focused on formation control and numerous approaches have been given to derive successful solutions to this [1, 5, 6]. The most commonly used approaches from the literature are roughly categorised as generic approaches, namely, generalised coordinates, behaviour-based, leader-follower strategy, virtual structure, and social potential field [10].

For the case of this paper, a leader-follower strategy will be used for formation control of the flock. LbCS will be used to design a motion planner for the flock navigating in the workspace in the presence of obstacles [8, 17, 18]. The flock needs to coordinate its motion in a well planned manner to successfully converge to its pre-defined target, avoiding obstacles present in its path. In our case, the obstacles will be spherical fixed obstacles, and the moving aircraft in the swarm itself will become the obstacles for all the other members. Using artificial potential fields, the Direct Method of Lyapunov will then be used to derive continuous acceleration-based controllers, which will give our system stability. The work to be done in this paper is an extension of work published in [11, 12]. The remainder of this paper is structured as follows: in Sect. 2, the fixed-wing aircraft model is defined; in Sect. 3, the artificial potential field functions are defined; in Sect. 4, the acceleration-based control laws are derived, while in Sect. 5, stability analysis of the robotic system is carried out; in Sect. 6, we demonstrate the effectiveness of the proposed controllers via a computer simulation; and finally, Sect. 7 concludes the paper and outlines the future work to be done in the area.

2 Vehicle Model

In this section, we derive a simple kinematic model for the leader-follower based formation control of multiple fixed-wing aircraft. We will consider n , $n \in \mathbb{N}$, fixed-wing aircraft in the Euclidean plane. We let \mathcal{A}_1 represent the leader and \mathcal{A}_i for $i = 2, \dots, n$ take the role of follower robots. We let (x_i, y_i) represents the Cartesian coordinates and gives the reference point in the 2-dimensional space, of our fixed-wing aircraft as seen in Fig. 1. Further more, let θ_i be the orientation of aircraft with respect to the z_1 -axis, also let L be the overall length and l be the total wingspan of the aircraft. We ignore the rolling and pitch angles. The aircraft safely needs to steer past an obstacles and to ensure this, the nomenclature of [15] is adopted and the spherical regions protecting the aircraft is constructed. Given the clearance parameters $\epsilon_1 > 0$ and $\epsilon_2 > 0$ and, we enclose the aircraft by a protective spherical region centered at (x_i, y_i, z_i) , with radius

$$r_i = \frac{\sqrt{(L + 2\epsilon_1)^2 + (l + 2\epsilon_2)^2}}{2} \tag{1}$$

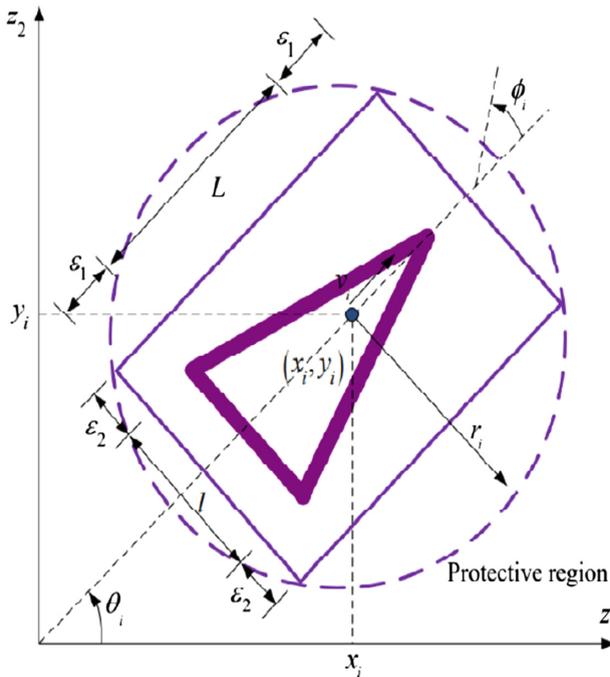


Fig. 1. Kinematic model of the fixed-wing UAV

The model of the system, adopted from [3] is

$$\begin{aligned}
 \dot{x}_i &= v_i \cos \theta_i - \frac{L}{2} \omega_i \sin \theta_i, \\
 \dot{y}_i &= v_i \sin \theta_i + \frac{L}{2} \omega_i \cos \theta_i, \\
 \dot{z}_i &= \nu_i, \\
 \dot{\theta}_i &= \omega_i, \\
 \dot{v}_i &= \sigma_i 1, \\
 \dot{\omega}_i &= \sigma_i 2, \\
 \dot{\nu}_i &= \sigma_i 3.
 \end{aligned} \tag{2}$$

From the above model of the system, v_i represents the instantaneous translational velocity of the aircraft, ω_i represents the yaw rate, ν_i represents the rate of altitude change, and $\sigma_{i1}, \sigma_{i2}, \sigma_{i3}$ for $i = 1, \dots, 3$ are the instantaneous accelerations. Now, system (2) is a description of the instantaneous velocities and accelerations of the fixed-wing aircraft. The state of the fixed-wing UAV can then be described by $\mathbf{x} = (x_i, y_i, z_i, \theta_i, v_i, \omega_i, \nu_i) \in \mathbb{R}^7$

3 Artificial Potential Field Function

Here we formulate collision free trajectories of the fixed-wing aircraft system under kinodynamic constraints in a given workspace.

3.1 Attractive Potential Field Functions

Attraction to Target. Leader is assigned a stationary target here and we let the target be a sphere with center (p_{11}, p_{12}, p_{13}) and radius r_t . The follower vehicles move with its leader maintaining desired formation and splits in case of danger and obstacles. We consider an attractive potential function for the attraction of the leader to its target:

$$V_1(\mathbf{x}) = \frac{1}{2}[(x_1 - p_{11})^2 + (y_1 - p_{12})^2 + (z_1 - p_{13})^2 + v_1^2 + \omega_1^2 + \nu_1^2]. \tag{3}$$

For the follower vehicles, $i = 2, \dots, n$ to maintain its desired relative position with respect to the leader whose center is given by $(p_{i1}, p_{i2}, p_{i3}) = (x_{11} - a_i, y_{11} - b_i, z_{11} - c_i)$ where a_i, b_i and c_i are relative x, y, z distance of the followers from the leader, we define the following function

$$V_i(\mathbf{x}) = \frac{1}{2}[(x_i - (x_i - a_i))^2 + (y_i - (y_i - b_i))^2 + (z_i - (z_i - c_i))^2 + v_i^2 + \omega_i^2 + \nu_i^2]. \tag{4}$$

Auxiliary Function. We design an auxiliary function to guarantee convergence of the leader and its designed in the form of

$$G_1(\mathbf{x}) = \frac{1}{2}[(x_1 - p_{11})^2 + (y_1 - p_{12})^2 + (z_1 - p_{13})^2]. \tag{5}$$

where p_{14} is the final orientation of the leader robots. For the follower vehicles $i = 2, \dots, n$, we have

$$G_i(\mathbf{x}) = \frac{1}{2}[(x_i - p_{i1})^2 + (y_i - p_{i2})^2 + (z_i - p_{i3})^2]. \tag{6}$$

The auxiliary functions guarantees controllers becomes zero at the leaders target. These functions above is later multiplied to the repulsive potential functions given in the following functions.

3.2 Repulsive Potential Field Functions

Unknown agents such as obstacles are always present in the environment and we design obstacles avoidance function by measuring Euclidean distance between the robots and obstacles present in our path. This will ensure all these obstacles will be avoided during navigation.

Fixed Obstacles in the Workspace. Lets fix $q \in \mathbb{N}$ solid obstacles within the boundaries of the workspace. We assume the l th obstacle be a sphere with center (o_{l1}, o_{l2}, o_{l3}) and radius ro_l . For the aircraft to avoid the l th obstacle, we consider the avoidance function in the form of

$$FO_{il}(\mathbf{x}) = \frac{1}{2}[(x_i - o_{l1})^2 + (y_i - o_{l2})^2 + (z_i - o_{l3})^2 - (ro_l + r_i)^2]. \tag{7}$$

where $r = 1, \dots, q$ and $i = 1, \dots, n$.

Moving Obstacles in the Workspace. Aircrafts in multi-agent system itself becomes obstacles for all the other members and it needs to taken care off to avoid collision. Therefore, for vehicle B_i to avoid vehicle B_j we design repulsive potential field function of the form.

$$MO_{ij}(\mathbf{x}) = \frac{1}{2}[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 - (2r_i)^2]. \tag{8}$$

for $i, j = 1, \dots, n, j \neq i$.

These obstacle avoidance functions and other functions given in the following sections are then combined with appropriate tuning parameters to generate repulsive potential field functions in the workspace.

Workspace Limitations. We desire to setup a definite 3D framework of dimension η_1 by η_2 by η_3 for the workspace of our aircraft. These boundaries are considered as *fixed obstacles* in our LbCS. We define the following potential function for the aircraft to avoid these:

$$\begin{aligned}
 W_{i1}(\mathbf{x}) &= (x_i - r_i), \\
 W_{i2}(\mathbf{x}) &= (\eta_2 - (y_i + r_i)), \\
 W_{i3}(\mathbf{x}) &= (\eta_1 - (x_i + r_i)), \\
 W_{i4}(\mathbf{x}) &= (y_i - r_i), \\
 W_{i5}(\mathbf{x}) &= (z_i - r_i), \\
 W_{i6}(\mathbf{x}) &= (\eta_3 - (z_i + r_i)).
 \end{aligned} \tag{9}$$

for $i = 1, \dots, n$.

Dynamic Constraints. Practically, there are yaw angle restrictions and aircraft translational speeds are restricted due to safety reasons. Subsequently, we have:

- (i) $v_{min} < v_i < v_{max}$, where v_{min} is the *minimum speed* for the aircraft while v_{max} is the *maximal achievable speed* of the aircraft,
- (ii) $|\omega_i| < \omega_{max}$, where ω_{max} is the maximum yaw rate,
- (iii) $|\nu_i| < \nu_{max}$, where ν_{max} is the maximum altitude change.

Hence, we consider the following avoidance functions:

$$\begin{aligned}
 U_{i1}(\mathbf{x}) &= \frac{1}{2}[(v_{max} - v_i)(v_{max} + v_i)], \\
 U_{i2}(\mathbf{x}) &= \frac{1}{2}[(v_i - v_{min})(v_i + v_{min})], \\
 U_{i3}(\mathbf{x}) &= \frac{1}{2}[\omega_{max} - \omega_i)(\omega_{max} + \omega_i)], \\
 U_{i4}(\mathbf{x}) &= \frac{1}{2}[(\nu_{max} - \nu_i)(\nu_{max} + \nu_i)].
 \end{aligned} \tag{10}$$

$i = 1, \dots, n$. The above positive functions guarantees the adherence to the limitations imposed upon the yaw angle and the velocities of the aircraft when encoded appropriately into the Lyapunov function.

4 Design of the Acceleration Controllers

Here we design nonlinear acceleration control laws for the system (2) using LbCS as proposed in [15].

4.1 Lyapunov Function

Using the above attractive and repulsive potential field functions we now construct the total potentials, that is, a Lyapunov function for system (1). We introduce the following control parameters for $i = 1, \dots, n$ that we will use in the repulsive potential functions.

- (i) $\alpha_{il} > 0, l = 1, \dots, q$, for avoidance of q sphere-like obstacles.
- (ii) $\beta_{ik} > 0, k = 1, \dots, 6$, for workspace of aircraft. These are treated as fixed obstacles.
- (iii) $\gamma_{is} > 0, s = 1, \dots, 4$, for avoidance of the artificial obstacles from dynamic constraints.
- (iv) $\eta_{ij} > 0, j = 1, \dots, n, j \neq i$, for the collision avoidance between any two aircraft.

$$L(\mathbf{x}) = \sum_{i=1}^n \left[V_i(\mathbf{x}) + \mathbf{G}_i(\mathbf{x}) \left(\sum_{l=1}^q \frac{\alpha_{il}}{\mathbf{FO}_{il}(\mathbf{x})} + \sum_{k=1}^6 \frac{\beta_{ik}}{\mathbf{W}_{ik}(\mathbf{x})} + \sum_{s=1}^4 \frac{\gamma_{is}}{\mathbf{U}_{is}(\mathbf{x})} + \sum_{\substack{j=1, \\ j \neq i}}^n \frac{\eta_{ij}}{\mathbf{M}_{ij}(\mathbf{x})} \right) \right]. \quad (11)$$

4.2 Nonlinear Acceleration Controllers

The design of the feedback controllers begins by noting that the functions f_i for $i = 1, \dots, 4$ are defined as (on suppressing \mathbf{x}):

$$\begin{aligned} f_{11} = & \left[1 + \sum_{l=1}^q \frac{\alpha_{1l}}{FO_{1l}} + \sum_{k=1}^6 \frac{\beta_{1k}}{W_{1k}} + \sum_{s=1}^4 \frac{\gamma_{1s}}{U_{1s}} + \sum_{j=2}^n \frac{\eta_{1j}}{M_{1j}} \right] (x_1 - p_{11}) - \frac{G_1 \beta_{11}}{(W_{11})^2} + \frac{G_1 \beta_{11}}{(W_{13})^2} \\ & - \sum_{i=2}^n \left[1 + \sum_{l=1}^q \frac{\alpha_{il}}{FO_{il}} + \sum_{\substack{j=1, \\ j \neq i}}^n \frac{\eta_{ij}}{M_{ij}} + \sum_{s=1}^4 \frac{\gamma_{1s}}{U_{1s}} \right] (x_i - (x_1 - a_i)) \\ & - G_1 \left[\sum_{l=1}^q \frac{\alpha_{1l}}{(FO_{1l})^2} (y_1 - o_{1l}) - 2 \sum_{j=2}^n \frac{\eta_{1j}}{(M_{1j})^2} (x_1 - x_j) \right], \\ f_{12} = & \left[1 + \sum_{l=1}^q \frac{\alpha_{1l}}{FO_{12}} + \sum_{k=1}^6 \frac{\beta_{1k}}{W_{1k}} + \sum_{s=1}^4 \frac{\gamma_{1s}}{U_{1s}} + \sum_{j=2}^n \frac{\eta_{1j}}{M_{1j}} \right] (y_1 - p_{12}) - \frac{G_1 \beta_{11}}{(W_{12})^2} + \frac{G_1 \beta_{11}}{(W_{14})^2} \\ & - \sum_{i=2}^n \left[1 + \sum_{l=1}^q \frac{\alpha_{il}}{(FO_{il})} + \sum_{\substack{j=1, \\ j \neq i}}^n \frac{\eta_{ij}}{M_{ij}} + \sum_{s=1}^4 \frac{\gamma_{1s}}{U_{1s}} \right] (y_i - (y_1 - b_i)) \\ & - G_1 \left[\sum_{l=1}^q \frac{\alpha_{1l}}{(FO_{1l})^2} (y_1 - o_{1l}) - 2 \sum_{j=2}^n \frac{\eta_{1j}}{(M_{1j})^2} (y_1 - y_j) \right], \\ f_{13} = & \left[1 + \sum_{l=1}^q \frac{\alpha_{1l}}{FO_{13}} + \sum_{k=1}^6 \frac{\beta_{1k}}{W_{1k}} + \sum_{s=1}^4 \frac{\gamma_{1s}}{U_{1s}} + \sum_{j=2}^n \frac{\eta_{1j}}{M_{1j}} \right] (z_1 - p_{13}) - \frac{G_1 \beta_{11}}{(W_{15})^2} + \frac{G_1 \beta_{11}}{(W_{16})^2} \\ & - \sum_{i=2}^n \left[1 + \sum_{l=1}^q \frac{\alpha_{il}}{(FO_{il})} + \sum_{\substack{j=1, \\ j \neq i}}^n \frac{\eta_{ij}}{M_{ij}} + \sum_{s=1}^4 \frac{\gamma_{1s}}{U_{1s}} \right] (z_i - (z_1 - c_i)) \\ & - G_1 \left[\sum_{l=1}^q \frac{\alpha_{1l}}{(FO_{1l})^2} (z_1 - o_{1l}) - 2 \sum_{j=2}^n \frac{\eta_{1j}}{(M_{1j})^2} (z_1 - z_j) \right], \end{aligned}$$

$$\begin{aligned}
 f_{14} &= 1 + G_1 \left[\frac{\gamma_{11}}{(U_{11})^2} - \frac{\gamma_{12}}{(U_{12})^2} \right], \\
 f_{15} &= 1 + \frac{\gamma_{13}G_1}{(U_{13})^2}, \\
 f_{16} &= 1 + \frac{\gamma_{14}G_1}{(U_{14})^2}.
 \end{aligned}$$

and for $i = 2, \dots, n$

$$\begin{aligned}
 f_{i1} &= \left[1 + \sum_{l=1}^q \frac{\alpha_{il}}{FO_{il}} + \sum_{k=1}^6 \frac{\beta_{ik}}{W_{ik}} + \sum_{s=1}^4 \frac{\gamma_{is}}{U_{is}} + \sum_{j=1}^n \frac{\eta_{ij}}{M_{ij}} \right] (x_i - (x_1 - a_i)) - \frac{G_i \beta_{i1}}{(W_{i1})^2} \\
 &\quad + \frac{G_i \beta_{i1}}{(W_{i3})^2} - G_i \left[\sum_{l=1}^q \frac{\alpha_{il}}{(FO_{il})^2} (x_i - o_{l1}) + 2 \sum_{j=1}^n \frac{\eta_{ij}}{(M_{ij})^2} (x_i - x_j) \right], \\
 f_{i2} &= \left[1 + \sum_{l=1}^q \frac{\alpha_{il}}{FO_{il}} + \sum_{k=1}^6 \frac{\beta_{ik}}{W_{ik}} + \sum_{s=1}^4 \frac{\gamma_{is}}{U_{is}} + \sum_{j=1}^n \frac{\eta_{ij}}{M_{ij}} \right] (y_i - (y_1 - b_i)) - \frac{G_i \beta_{i2}}{(W_{i2})^2} \\
 &\quad + \frac{G_i \beta_{i4}}{(W_{i4})^2} - G_i \left[\sum_{l=1}^q \frac{\alpha_{il}}{(FO_{il})^2} (y_i - o_{l2}) + 2 \sum_{j=1}^n \frac{\eta_{ij}}{(M_{ij})^2} (y_i - y_j) \right], \\
 f_{i3} &= \left[1 + \sum_{l=1}^q \frac{\alpha_{il}}{FO_{il}} + \sum_{k=1}^6 \frac{\beta_{ik}}{W_{ik}} + \sum_{s=1}^4 \frac{\gamma_{is}}{U_{is}} + \sum_{j=1}^n \frac{\eta_{ij}}{M_{ij}} \right] (z_i - (z_1 - c_i)) - \frac{G_i \beta_{i5}}{(W_{i5})^2} \\
 &\quad + \frac{G_i \beta_{i6}}{(W_{i6})^2} - G_i \left[\sum_{l=1}^q \frac{\alpha_{il}}{(FO_{il})^2} (z_i - o_{l3}) + 2 \sum_{j=1}^n \frac{\eta_{ij}}{(M_{ij})^2} (z_i - z_j) \right], \\
 f_{i4} &= 1 + G_i \left[\frac{\gamma_{i1}}{(U_{i1})^2} - \frac{\gamma_{i2}}{(U_{i2})^2} \right], \\
 f_{i5} &= 1 + \frac{\gamma_{i3}G_i}{(U_{i3})^2}, \\
 f_{i6} &= 1 + \frac{\gamma_{i4}G_i}{(U_{i4})^2}.
 \end{aligned}$$

So, we design the following theorem:

Theorem 1. Consider n mobile robots in the C-ITS whose motion are governed by the ODE's described in system (2). The principal goal is to facilitate maneuvers within a constrained environment and reach the target configuration. The subtasks include; restrictions placed on the workspace, convergence to predefined targets, and consideration of kinodynamic constraints. Utilizing the attractive and repulsive potential field functions, the following continuous time-invariant acceleration control laws can be generated, that intrinsically guarantees stability, in the sense of Lyapunov, of system (2) as well:

$$\begin{aligned}
 \sigma_{i1} &= \frac{-(\delta_{i1}v_i + f_{i1}\cos\theta_i + f_{i2}\sin\theta_i)}{f_{i4}}, \\
 \sigma_{i2} &= \frac{-(\delta_{i2}\omega_i + \frac{L}{2}f_{i2}\cos\theta_i - \frac{L}{2}f_{i1}\sin\theta_i)}{f_{i5}}, \\
 \sigma_{i3} &= \frac{-(\delta_{i3}\nu_i + f_{i3})}{f_{i6}}.
 \end{aligned} \tag{12}$$

where $\delta_i > 0$, for $i = 1, \dots, 3$ are constants commonly known as convergence parameters.

5 Stability Analysis

Theorem 2. *Given that $\mathbf{x}_e := (p_{i1}, p_{i2}, p_{i3}, p_{i4}, 0, 0, 0) \in \mathbb{R}^7$, is an equilibrium point for (2), then $\mathbf{x}_e \in D(L(\mathbf{x}))$ is a stable equilibrium point of system (2).*

Proof. One can easily verify the following:

1. $L(\mathbf{x})$ is defined, continuous and positive over the domain $D(L_2(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^7 : FO_{il}(\mathbf{x}) > 0, l = 1, \dots, q, MO_{ij}(\mathbf{x}) > 0, i, j = 1, \dots, n, i \neq j, U_{is}(\mathbf{x}) > 0, s = 1, \dots, 4, W_{ik}(\mathbf{x}) > 0, k = 1, \dots, 6;$
2. $L(\mathbf{x}_e) = 0;$
3. $L(\mathbf{x}) > 0 \ \forall \mathbf{x} \in D(L(\mathbf{x}))/\mathbf{x}_e.$

Next, consider the time derivative of the candidate Lyapunov function along a particular trajectory of system (2):

$$\dot{L}_2(\mathbf{x}) = \sum_{i=1}^n [f_{i1}\dot{x}_i + f_{i2}\dot{y}_i + f_{i3}\dot{z}_i + f_{i4}v_i\dot{v}_i + f_{i5}\omega_i\dot{\omega}_i + f_{i6}\nu_i\dot{\nu}_i].$$

Substituting the controllers given in (12) and the governing ODEs for system (2), we obtain the following semi-negative definite function

$$\dot{L}_2(\mathbf{x}) = -(\delta_{i1}v_i^2 + \delta_{i2}\omega_i^2 + \delta_{i3}\nu_i^2) \leq 0.$$

Thus, $\dot{L}_2(\mathbf{x}) \leq 0 \ \forall \mathbf{x} \in D(L(\mathbf{x}))$ and $\dot{L}_2(\mathbf{x}_e) = 0$. Finally, it can be easily verified that $L(\mathbf{x}) \in C^1(D(L(\mathbf{x})))$, which makes up the fifth and final criterion of a Lyapunov function. Hence, $L(\mathbf{x})$ is classified as a Lyapunov function for system (2) and \mathbf{x}_e is a stable equilibrium point in the sense of Lyapunov. \square

6 Simulation Results and Discussion

6.1 Scenario 1: Arrowhead Formation in the Presence of Obstacles

We consider the motion of four vehicles in an arrowhead formation in the presence of obstacles. The follower vehicles are prescribed a position relative to the leader. While the leader moves towards its target, the follower maintains a desired distance and orientation relative to the leader. In the presence of obstacles, the formation splits and rejoins at safer places throughout the journey, as given in Fig. 2. Clearly, the vehicles also avoid inter-collision, which is treated as moving obstacles. Corresponding initial and final configurations together with the other necessary parameters required in the simulation are given in Table 1, assuming all the values have been accounted for. The evolution of the Lyapunov function (red) together with its derivative (blue) along the system trajectory is given in Fig. 4, which shows the system finally stops at the target since the energy of the system becomes zero. Translational and rotational accelerations given in Fig. 6 clearly decrease in the presence of obstacles, and later increase after avoiding the obstacles. The acceleration of the system finally converges to zero as it approaches the target.

Table 1. Numerical values of initial and final states, constraint and parameters for Scenario 1 given in Fig. 2

	Initial configuration
Workspace	$\eta_1 = 2000, \eta_2 = 500, \eta_3 = 500$
Rectangular position	$(x_1, y_1, z_1) = (100, 200, 250), (x_2, y_2, z_2) = (100, 100, 250)$ $(x_3, y_3, z_3) = (100, 300, 250), (x_4, y_4, z_4) = (100, 400, 250)$
Velocities	$v_i = 1.5, \omega_i = 0.5, \nu_i = 0.5$ for $i = 1, \dots, n$,
Angular positions	$\theta_i = 0$ for $i = 1, \dots, n$,
	Constraints and parameters
Dimensions of UAV	$L = 20, w = 10$
Leaders target	$(p_{11}, p_{12}, p_{13}) = (1900, 200, 150), rt_1 = 20$
Position of ghost targets	$(a_1, b_1, c_1) = (60, -80, 0), (a_2, b_2, c_2) = (60, 0, 0),$ $(a_3, b_3, c_3) = (60, 80, 0),$
Fixed obstacles	$(o_{11}, o_{12}, o_{13}) = (700, 280, 240), (o_{21}, o_{22}, o_{23}) = (1500, 150, 110),$ $(o_{31}, o_{32}, o_{33}) = (1100, 100, 200)$
Velocity constraints	$v_{\max} = 5, v_{\min} = 1, \omega_{\max} = 1, \nu_{\max} = 1$
Clearance parameters	$\epsilon_1 = 0.1, \epsilon_2 = 0.05$
	Control and convergence parameters
Collision avoidance	$\alpha_{il} = 0.1$, for $l = 1, \dots, 5, \beta_{ik} = 0.1$ for $k = 1, \dots, 6$
Dynamics constraints	$\gamma_s = 0.0001$, for $s = 1, \dots, 4$
Convergence	$\delta_{1k} = 4000$, for $k = 1, \dots, 3,$ $\delta_{ik} = 100$, for $i = 2, \dots, 4$ and $k = 1, \dots, 3$

6.2 Scenario 2: Double Platoon Formation in the Presence of Obstacles

The motion of four vehicles in a double platoon formation in the presence of obstacles is considered in this scenario. The follower vehicles are prescribed a position relative to the leader. The leader moves towards its pre-defined target and the followers maintain a desired distance and orientation. In the presence of obstacles, the flock splits and rejoins at a safer distance throughout the journey, as given in Fig. 3. Clearly, the vehicles also avoid inter-collisions, which are treated as moving obstacles. Corresponding initial and final configurations together with the other necessary parameters required in the simulation are given in Table 2, assuming all the values have been accounted for. The evolution of the

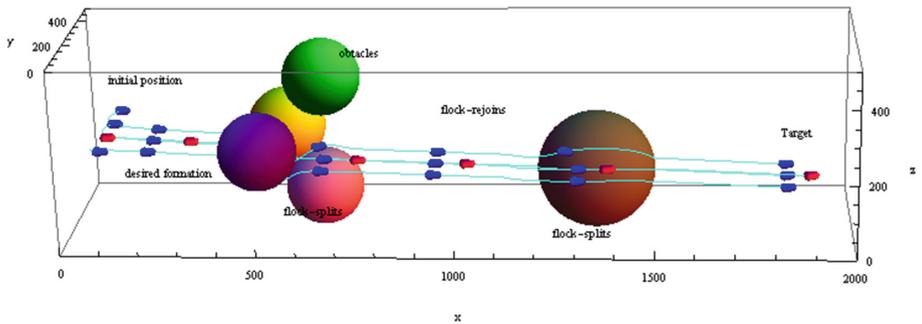


Fig. 2. The evolution of trajectories of 3D multiple fixed-wing UAVs in arrowhead formation. The leader is in red and the followers are in blue. The cyan lines shows the trajectories in presence of multiple spherical obstacles.

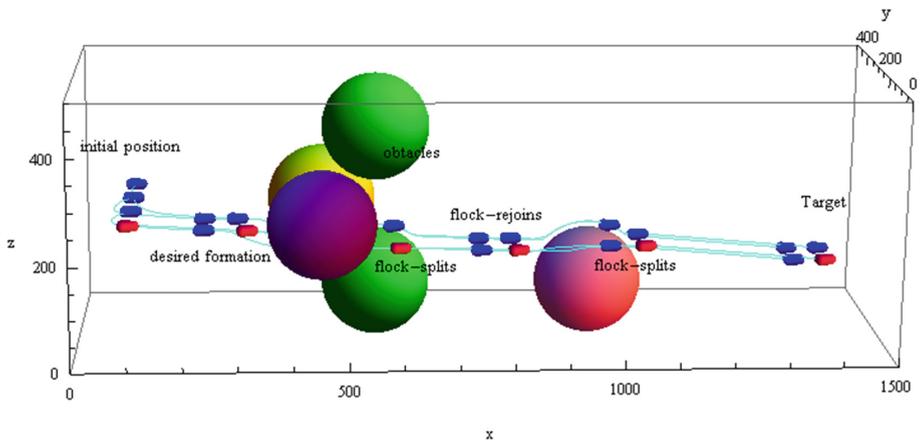


Fig. 3. The evolution of trajectories of 3D multiple fixed-wing UAVs for double platoon formation. The leader is in red and the followers are in blue. The cyan lines shows the trajectories in presence of multiple spherical.

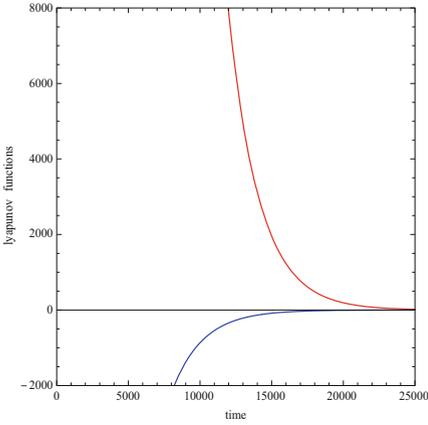


Fig. 4. Evolution of $L(\mathbf{x})$ in red and $\dot{L}_{(2)}(\mathbf{x})$ in blue for scenario 1.

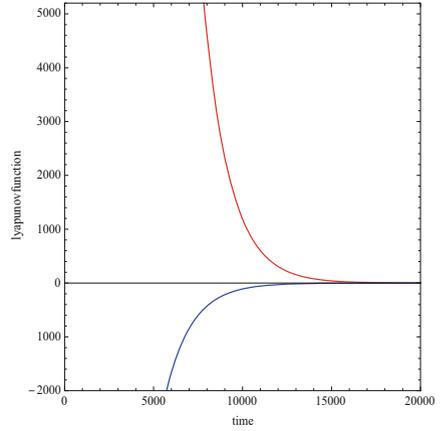


Fig. 5. Evolution of $L(\mathbf{x})$ in red and $\dot{L}_{(2)}(\mathbf{x})$ in blue for scenario 2.

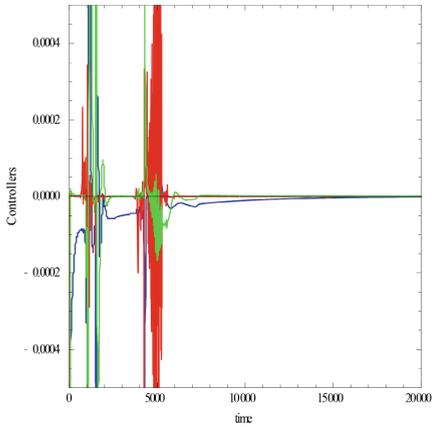


Fig. 6. Evolution of acceleration σ_{11} in blue, σ_{12} in red and σ_{13} in green for scenario 1.

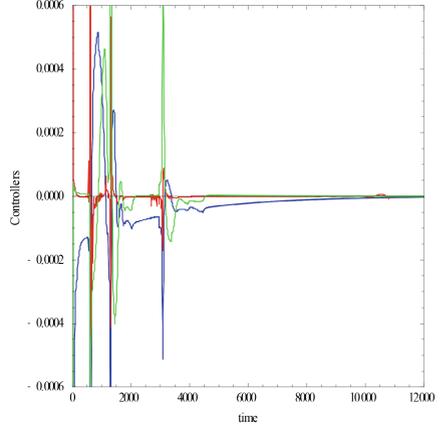


Fig. 7. Evolution of acceleration σ_{11} in blue, σ_{12} in red and σ_{13} in green for scenario 2.

Lyapunov function (red) together with its derivative (blue) along the system trajectory is given in Fig. 5, which shows the system finally stops at the target since the energy of the system becomes zero. Translational and rotational accelerations given in Fig. 7 clearly decrease in cases of obstacles, and later increase after avoiding the obstacles. The acceleration of the system finally converges to zero as it approaches the target.

Table 2. Numerical values of initial and final states, constraint and parameters for Scenario 2 given in Fig. 3

	Initial configuration
Workspace	$\eta_1 = 2000, \eta_2 = 500, \eta_3 = 500$
Rectangular positions	$(x_1, y_1, z_1) = (100, 100, 250), (x_2, y_2, z_2) = (100, 200, 250)$ $(x_3, y_3, z_3) = (100, 300, 250), (x_4, y_4, z_4) = (100, 400, 250)$
Velocities	$v_i = 1.5, \omega_i = 0.5, \nu_i = 0.5$ for $i = 1, \dots, n$,
Angular positions	$\theta_i = 0$ for $i = 1, \dots, n$,
	Constraints and parameters
Dimensions of UAV	$L = 20, l = 10$
Leaders target	$(p_{11}, p_{12}, p_{13}) = (1900, 200, 150), rt_1 = 20$
Position of ghost targets	$(a_1, b_1, c_1) = (0, -80, 0), (a_2, b_2, c_2) = (60, -80, 0),$ $(a_3, b_3, c_3) = (60, 0, 0),$
Fixed obstacles	$(o_{11}, o_{12}, o_{13}) = (400, 250, 200), (o_{21}, o_{22}, o_{23}) = (1100, 200, 160),$ $(o_{31}, o_{32}, o_{33}) = (800, 150, 200),$
Velocity constraints	$v_{\max} = 5, v_{\min} = 1, \omega_{\max} = 1, \nu_{\max} = 1$
Clearance parameters	$\epsilon_1 = 0.1, \epsilon_2 = 0.05$
	Control and convergence parameters
Collision avoidance	$\alpha_{il} = 0.1$, for $l = 1, \dots, 5, \beta_{ik} = 0.1$ for $k = 1, \dots, 6$
Dynamics constraints	$\gamma_s = 0.0001$, for $s = 1, \dots, 4$
Convergence	$\delta_{1k} = 2800$, for $k = 1, \dots, 3$ $\delta_{ik} = 50$, for $i = 2, \dots, 4$ and $k = 1, \dots, 3$

7 Conclusion

In conclusion, this paper addresses the solution of a motion planner for multiple fixed-wing aircraft in an environment populated with obstacles using the Lyapunov-based Control Scheme. The strategy that has been used in this paper, is the leader-follower approach via split rejoin maneuvers. In this approach, a leader was defined, and it took the responsibility of the leader. The remaining robots in the formation became followers and they positioned themselves relative to the leader aircraft. The followers were required to position and follow the leader's pre-defined distance. The formation navigated in the environment in the pre-defined formation, moving towards its predefined target. In the case of unknown agents, such as obstacles, the flock splits and rejoins at a safer place or distance. The formation also avoided inter-collision, which was treated as a moving obstacle in this case. The controllers derived using the Lyapunov-based Control Scheme produced feasible trajectories and also ensured a nice convergence of the system to its equilibrium system, satisfying the necessary constraints. The controllers slowed down and decreased in case of obstacles and increased later, approaching zero near targets. Finally, computer simulations of different scenarios have been used to demonstrate the control laws and show their effectiveness. Various graphs, of which some are present in the discussion, have been generated

and they clearly show that this is one of the most powerful techniques in proving the stability of a non-linear system. Future work includes the introduction of obstacles such as rectangular planes and cylindrical obstacles. One can also look at the dynamic analysis of stability by applying the leader-follower approach to other mechanical robots such as quadrotors. Most importantly, proving the achieved results practically would be an interesting thing to work with.

References

1. Gazi, V.: Swarm aggregations using artificial potentials and sliding-mode control. *IEEE Trans. Rob.* **21**(6), 1208–1214 (2005). <https://doi.org/10.1109/TRO.2005.853487>
2. Goh, G.D., Agarwala, S., Goh, G., Dikshit, V., Sing, S.L., Yeong, W.Y.: Additive manufacturing in unmanned aerial vehicles (UAVs): challenges and potential. *Aerosp. Sci. Technol.* **63**, 140–151 (2017). <https://doi.org/10.1016/j.ast.2016.12.019>
3. Hwangbo, M., Kuffner, J., Kanade, T.: Efficient two-phase 3d motion planning for small fixed-wing UAVs. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1035–1041. IEEE (2007). <https://doi.org/10.1109/ROBOT.2007.363121>
4. Kan, X., Thomas, J., Teng, H., Tanner, H.G., Kumar, V., Karydis, K.: Analysis of ground effect for small-scale UAVs in forward flight. *IEEE Robot. Autom. Lett.* **4**(4), 3860–3867 (2019). <https://doi.org/10.1109/LRA.2019.2929993>
5. Kumar, D., Raj, J., Raghuwaiya, K., Vanualailai, J.: Autonomous UAV landing on mobile platforms. In: *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pp. 1–6. IEEE (2021). <https://doi.org/10.1109/CSDE53843.2021.9718368>
6. Kumar, S.A., Vanualailai, J., Sharma, B., Prasad, A.: Velocity controllers for a swarm of unmanned aerial vehicles. *J. Ind. Inf. Integr.* **22**, 100198 (2021). <https://doi.org/10.1016/j.jii.2020.100198>
7. Kumar, S.A., Vanualailai, J., Sharma, B.: Lyapunov-based control for a swarm of planar nonholonomic vehicles. *Math. Comput. Sci.* **9**(4), 461–475 (2015). <https://doi.org/10.1109/CCA.2015.7320890>
8. Mamino, M., Viglietta, G.: Square formation by asynchronous oblivious robots. *arXiv preprint arXiv:1605.06093* (2016). <https://doi.org/10.48550/arXiv.1605.06093>
9. Quintero, S.A., Collins, G.E., Hespanha, J.P.: Flocking with fixed-wing UAVs for distributed sensing: a stochastic optimal control approach. In: *2013 American Control Conference*, pp. 2025–2031. IEEE (2013). <https://doi.org/10.1109/ACC.2013.6580133>
10. Raghuwaiya, K.: Stability of swarm-like cooperative systems. Ph.D. thesis, University of the South Pacific, Suva, Fiji Islands, August 2016. Ph.D. dissertation
11. Raghuwaiya, K., Chand, R.: 3d motion planning of a fixed-wing unmanned aerial vehicle. In: *2018 5th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pp. 241–245. IEEE (2018). <https://doi.org/10.1109/APWC on CSE.2018.00046>
12. Raghuwaiya, K., Sharma, B., Vanualailai, J.: Leader-follower based locally rigid formation control. *J. Adv. Transp.* **2018** (2018). <https://doi.org/10.1155/2018/5278565>

13. Raj, J., Raghuwaiya, K., Sharma, B., Vanualailai, J.: Motion control of a flock of 1-trailer robots with swarm avoidance. *Robotica* **39**(11), 1926–1951 (2021)
14. Raj, J., Raghuwaiya, K.S., Vanualailai, J.: Novel Lyapunov-based autonomous controllers for quadrotors. *IEEE Access* **8**, 47393–47406 (2020). <https://doi.org/10.1109/ACCESS.2020.2979223>
15. Sharma, B.: New directions in the applications of the Lyapunov-based control scheme to the findpath problem. Ph.D. thesis, University of the South Pacific, Suva, Fiji Islands, July 2008. Ph.D. dissertation
16. Sharma, B.N., Vanualailai, J., Chand, U.: Flocking of multi-agents in constrained environments. *Eur. J. Pure Appl. Math.* **2**(3), 401–425 (2009). <https://ejpam.com/index.php/ejpam/article/view/263>
17. Shojaei, K.: Neural adaptive output feedback formation control of type (m, s) wheeled mobile robots. *IET Control Theory Appl.* **11**(4), 504–515 (2017). <https://doi.org/10.1049/iet-cta.2016.0952>
18. Vanualailai, J., Sharan, A., Sharma, B.: A swarm model for planar formations of multiple autonomous unmanned aerial vehicles. In: 2013 IEEE International Symposium on Intelligent Control (ISIC), pp. 206–211. IEEE (2013). <https://doi.org/10.1109/ISIC.2013.6658618>
19. Wang, X., et al.: Coordinated flight control of miniature fixed-wing UAV swarms: methods and experiments. *Sci. China Inf. Sci.* **62**(11), 1–17 (2019). <https://doi.org/10.1002/rnc.5030>
20. Zhang, J., Yan, J., Zhang, P.: Fixed-wing UAV formation control design with collision avoidance based on an improved artificial potential field. *IEEE Access* **6**, 78342–78351 (2018). <https://doi.org/10.1109/ACCESS.2018.2885003>



Prognostic Assessment of COVID-19 Vaccination Levels

Iryna Pikh^{1,2}, Vsevolod Senkivsky¹ , Alona Kudriashova¹  ,
and Nataliia Senkivska^{1,2} 

¹ Ukrainian Academy of Printing, Lviv, Ukraine
kudriashovaaliona@gmail.com

² Lviv Polytechnic National University, Lviv, Ukraine

Abstract. In order to move to a stable life rhythm and a satisfactory condition of people, which would ensure the organization of the usual mode of daily activities, it is necessary to achieve a sufficiently complete vaccination of the population in a region. At the same time, significant obstacles to achieving the desired result in Ukraine are the hesitation of a large part of the population regarding the vaccination, fear of a purely medical procedure, and distrust of its effectiveness. Due to the lack of a wide range of scientifically grounded research of this problem, insufficient attention is paid to a deeper analysis of the factors influencing the intensity and effectiveness of vaccination. In view of what has been said in the proposed article, many factors related to the vaccination process have been identified based on the developed ontology. A formalized representation of the connections between factors has been made using the semantic network as an information database, which has become a prerequisite for ranking by weight factors. Using the methodology of hierarchies modelling, the levels of factors preferences are established and a multilevel model of their priority influence on the researched process is synthesized. Alternative options for the vaccination process have been designed and a prognostic assessment of the levels of COVID-19 vaccination intensity has been carried out, which allows the selection of the optimal option for the specific parameters of the initial factors.

Keywords: COVID-19 · Vaccination · Ontology · Factors · Semantic network · Ranking · Multilevel model · Linear convolution criteria · Alternative option · Multi-criteria optimization · Membership function · Fuzzy relation · Convolution

1 Introduction

The threat of a pandemic due to COVID-19 emergence and its modifications (strains) in the global scale has emerged relatively recently, unexpectedly and unpredictably. The first efforts of doctors and health services were aimed at emergency care to reduce the number of deaths. General safety measures focused on isolating patients, minimizing contact and interaction between subjects through

“mechanical” means – wearing masks, using antiseptics, washing hands, safe distance. At the same time, considerable funds and efforts of scientists were directed to the development of special vaccines that would significantly protect people from infection. Vaccines of different levels have been created, statistics show a fairly high level of their effectiveness. It would seem that part of the problem has been solved to some extent: among the “new” patients admitted to medical institutions, the number of vaccinated ones compared to the total number of hospitalized is one to ten.

Salvation has been found, but not everything is so simple. As they say, a new problem has arisen “in problem-free area” – in some part of society, for unknown, often far-fetched, reasons, there is resistance to the vaccination process. There are always motives, but quite often sceptics are “convinced” of the need for vaccination by peak outbreaks in a particular region. It is clear that in this situation it is necessary to identify deeper causes and parameters (factors) of this phenomenon and to organize them in some way according to the importance of influencing the vaccination process in order to prognostically assess alternatives and choose the best vaccination option, which has prompted the authors to write this paper.

2 Problem Statement

To systematize the knowledge about COVID-19 vaccination and to identify influencing factors, an ontology is formed. The ontology structure directly affects the ability to establish the optimal solution of the main or secondary problems. The iterative approach to its creation consists in step-by-step studying (filling) of ontology. It is possible to constantly add new classes and connections between them, update and optimize the ontology.

In general, the construction of an ontological model involves: the accumulation of knowledge about the subject area; decomposition: division of the researched process into separate elements which will become a model basis; identification of elements; classification: definition of classes and elements belonging to them (hierarchy of classes); description of properties; assignment of property values; making connections; expansion and concretization of the ontology; verification; introduction of ontology [3,12,30] (Figs. 1 and 2).

Separation of factors related to the vaccination process is a necessary but insufficient condition for prognostic assessment of the vaccination levels. Further research is related to the establishment of ranks and weight values of factors, which leads to the use of appropriate theoretical principles - methods of hierarchical analysis, ranking, pairwise comparisons, graph theory, multilevel hierarchical systems.

A formalized description of the initial information area can be performed using the mechanism of semantic networks [21] and the means of synthesis of priority models of factors [11,14,24].

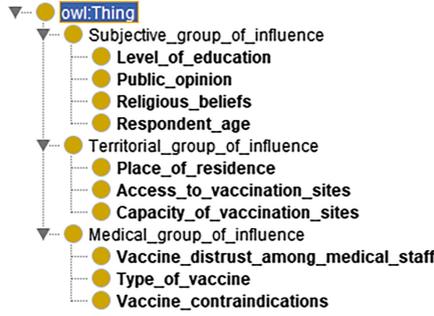


Fig. 1. Taxonomy of the ontology concepts

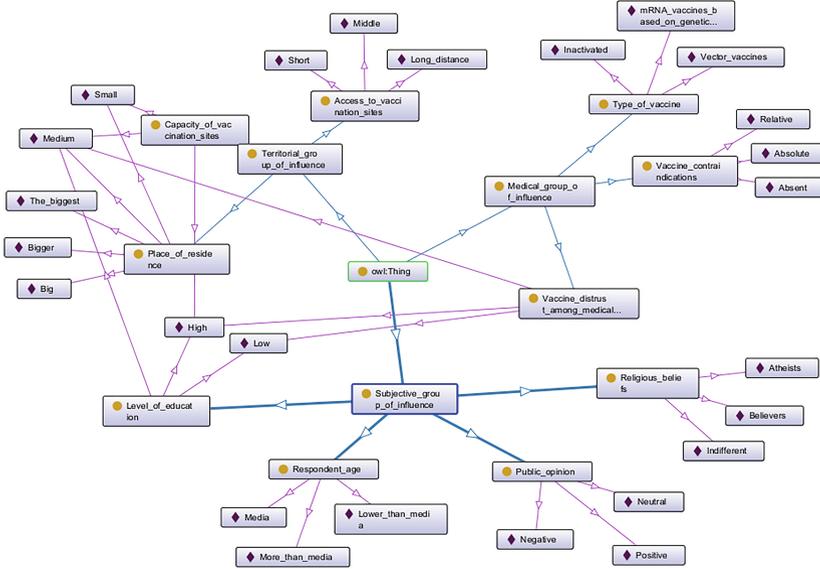


Fig. 2. General ontological graph of classes and samples

Based on the survey and the developed ontology, some set of factors is established related to COVID-19 vaccination, and written as a set:

$$X = \{x_1; x_2; \dots; x_n\}, \tag{1}$$

where: x_1 is the level of education (LE), x_2 is the public opinion (on the risk of coronavirus infection) (PO), x_3 are the religious beliefs (RB), x_4 is the respondent age (RA), x_5 is the place of residence (PR), x_6 is the access to vaccination sites (AV), x_7 is the capacity (of vaccination sites) (CV), x_8 is the distrust to medical stuff (DM), x_9 is the type of vaccine (TV), x_{10} is the vaccine contraindications (VC).

The nodes of the semantic network or its vertices will represent the semantics of terms – linguistic factors-arguments of the set (1) and the arc will represent pairs of vertices $(x_i, x_j), (i, j = 1 \div 10; i \neq j)$, for which the connection identified by the predicates is determined.

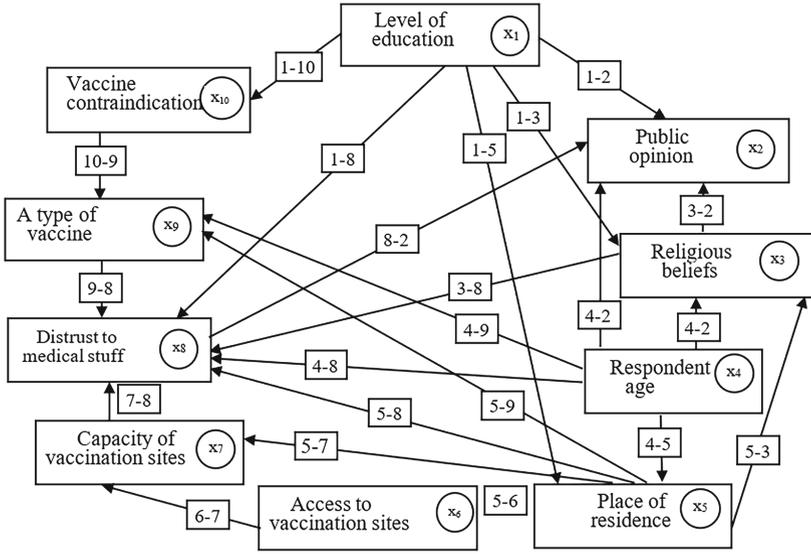


Fig. 3. Semantic network of factors influencing the vaccination intensity

The establishment of priority levels of factors is based on the method of mathematical modelling of hierarchies [11, 14, 25]. According to the method, the reachability matrix is formed - a mathematical analogue of the relationships between factors in the semantic network, according to the following rule:

$$b_{ij} = \begin{cases} 1, & \text{if from the vertex } i \text{ one can reach the vertex } j; \\ 0 & \text{in another case.} \end{cases} \quad (2)$$

A practical important continuation of the study is to establish the numerical weight of factors on the basis of pairwise comparisons and identify the predominance of one of them as a preference degree of the factor, obtaining a matrix of pairwise comparisons. This is a method of numerical or cardinal agreement on the priority level [15, 31], which can study not only the presence or absence of consistency in pairwise comparisons of factor weights, but also to obtain a numerical assessment of the adequacy of relationships between them in the original graphical representation.

The formation of alternative vaccination levels and determination of the optimal one can be done using the method of linear convolution of criteria, an effective means of which is considered to be a multi-criteria selection of alternative

[25], based on which a linear combination of partial functionals f_1, \dots, f_m comes into one generalized target functional:

$$F(w, x) = \sum_{i=1}^m w_i f_i(x) \rightarrow \max_{x \in D}; w \in W, \tag{3}$$

where $W = \{w = (w_1, \dots, w_m)^T; w_i > 0; \sum_{i=1}^m w_i = 1\}$.

The value of utility functions is equated to the corresponding weights of factors involved in the formation of alternatives, the optimal of which is determined by the theorem of the method of multi-criteria utility theory [23], the essence of which is that for conditionally independent in utility and preference criteria (factors) there is a utility function:

$$U(x) = \sum_{i=1}^m w_i u_i(y_i), \tag{4}$$

which serves as a benchmark for choosing the best option. Herewith $U(x)$ is the multi-criteria utility function ($0 \leq U(x) \leq 1$) of the alternative x ; $u_i(y_i)$ is the utility function of the i -th factor ($0 \leq u_i(y_i) \leq 1$); y_i is the value of the alternative x according to the factor i ; w_i is the weight of the i -th factor, moreover $0 < w_i < 1, \sum_{i=1}^m w_i = 1$.

The reliability of the study will be confirmed by the method of multi-criteria optimization [27], according to which the fuzzy relations of pairwise preferences between alternatives are defined. The essence of the method is to use the utility functions of alternatives. It is assumed that for clear utility functions f_j , the alternative x with higher rate $f_j(x)$ outweighs by the factor j the alternative y with lower rate $f_j(y)$ and this preference F_j is described by a clear relation:

$$F_j = \{(x, y) : f_j(x) \geq f_j(y), x, y \in X\}. \tag{5}$$

The condition of the optimal vaccination option will be the maximum value of the utility function of the alternative, taking into account the factors used for its design:

$$f_j(x_0) \geq f_j(y), \forall j = 1, m; \forall y \in X. \tag{6}$$

Finally, the Pareto-optimal alternative is obtained, which determines at which fuzzy relations of preferences between factors the highest level of vaccination intensity is received.

3 Literature Review

The research focused on the subject of the proposed article can be divided into certain groups, each of which is related to the general problems of the pandemic, the time stages of COVID-19 and its spread; treatment and protection measures against the infection, including vaccination; creation of vaccination means.

The first group of papers is devoted to the issues of COVID-19 detection, dynamics modelling, pandemic size in Ukraine and coronavirus mortality rates [13, 16, 20, 26]. A separate group includes publications focused on processing statistics on the pandemic spread and the development of certain patterns based on the results and forecasting the possible number of patients in a particular region for the specific period [4, 9, 22]. A significant number of publications are focused on studying the problem of vaccination, which addresses a wide range of issues – from understanding the problem and the importance of vaccination to modelling and analysing its influence on the spread of the coronavirus pandemic [1, 2, 5–8, 17–19, 28, 29]. The process of vaccination development is accompanied by informational publications on the vaccine from the World Health Organization and the European Medicines Agency [10].

At the same time, despite the availability of scientific and reference papers, the vaccination problem needs further study as one of the most effective measures to combat the COVID-19 pandemic. A brief analysis of the publications shows the lack of a theoretically grounded approach to the problem of forming the effectiveness of the vaccination process. The essence of scientific novelty and practical value of the solutions obtained in the paper is to use methods of systems analysis, operations research theory and hierarchy modelling as a basis for formalized expression of the influence degree of certain factors on the intensity and prognostic assessment of COVID-19 vaccination levels.

The Goal of the Research. The development of the ontology presented in the form of an ontological graph, the vertices of which are the components of the subject area, and the arcs indicate the direction of relations between concepts. The separation of many factors related to the COVID-19 vaccination process and a formalized representation of the connections between them through the semantic network. Ranking of factors according to the levels of priority influence on the studied process on the basis of calculated weight priorities. The formation of alternative options of vaccination process implementation and finding the optimal option for Pareto set factors by the method of linear convolution of criteria. The assessment of the reliability of the prognostically obtained vaccination level on the basis of simulation modelling, which is based on the method of multi-criteria optimization.

4 Material and Methods

The study of the vaccination intensity in a pandemic has been carried out on the basis of the above formalized representation of the connections between factors of the semantic network, followed by the method of mathematical modelling of hierarchies to rank factors according to the influence importance on the process; the method of linear convolution of criteria, which will provide an experimental design of alternative options for the implementation of the vaccination process and determine the best option; the method of multi-criteria optimization, software implementation of which will provide the result assessment.

Finally, the dependence of the vaccination intensity level on a set of factors influencing the studied process is reflected by the function:

$$Q = F(x_1; x_2; \dots; x_{10}). \tag{7}$$

Determining the weight of the function arguments (7) is possible on the basis of the binary reachability matrix formed by rule (2) (Table 1), which formalizes the connections between the factors specified in the semantic network.

Table 1. Reachability matrix

	X1 LE	X2 PO	X3 RB	X4 RA	X5 PR	X6 AV	X7 CV	X8 DM	X9 TV	X10 VC
X1, LE	1	1	1	0	1	0	0	1	0	1
X2, PO	0	1	0	0	0	0	0	0	0	0
X3, RB	0	1	1	0	0	0	0	1	0	0
X4, RA	0	1	1	1	1	0	0	1	1	0
X5, PR	0	0	0	0	1	1	1	1	1	0
X6, AV	0	0	0	0	0	1	1	0	0	0
X7, CV	0	0	0	0	0	0	1	1	0	0
X8, DM	0	1	0	0	0	0	0	1	0	0
X9, TV	0	0	0	0	0	0	0	1	1	0
X10, VC	0	0	0	0	0	0	0	0	1	1

Practically the vertex $x_j (j = 1, 2, \dots, 10)$ of the semantic network in Fig. 3 is considered achievable relative to the vertex $x_i (i = 1, 2, \dots, 10)$, if the latter can be reached in x_j . The result of the analysis of all vertices leads to a subset of achievable vertices $D(w_i)$. The opposite statement will be logical: one will consider the vertex x_i as the predecessor of the vertex x_j , if it is reached from it. The set of vertices of the predecessors forms a subset $P(w_i)$. The intersection of subsets of achievable vertices and predecessor vertices, i.e.:

$$Z(w_i) = D(w_i) \cap P(w_i), \tag{8}$$

the vertices of which are not reached from any of the vertices of the remaining set X , determines a certain priority level of the factors related to these vertices, provided:

$$P(w_i) = Z(w_i). \tag{9}$$

To determine the priority levels of factors, the reachability matrix and condition (8) and (9) is used, on the basis of which the first iterative table is constructed according to this rule. In the second column one enters a subset $D(w_i)$ - the numbers of achievable vertices, or the numbers of individual elements of the corresponding rows of the reachability matrix; the third column defines a subset of the vertices of the predecessors $P(w_i)$ - the numbers of the individual elements of this matrix columns. In this case, the dependence (8) will mean the

Table 2. xxx

i	$D(w_i)$	$P(w_i)$	$D(w_i) \cap P(w_i)$
1	1, 2, 3, 5, 8, 10	1	1
2	2	1, 2, 3, 4, 8	2
3	2, 3, 8	1, 3, 4	3
4	2, 3, 4, 5, 8, 9	4	4
5	5, 6, 7, 8, 9	1, 4, 5	5
6	6, 7	5, 6	6
7	7, 8	5, 6, 7	7
8	2, 8	1, 3, 4, 5, 7, 8, 9	8
9	8,9	4, 5, 9, 10	9
10	9,10	1,10	10

fulfilment of the equality condition of factor numbers in the second and third columns of the table, resulting in the formation of a certain level of factors.

As it can be seen from Table 2, the coincidence of numbers was recorded for factors 1 – level of education and 4 – respondent age. According to (8) and the general logic of the method used, these factors will be considered the highest in terms of the influence priority on the vaccination process.

According to the methods of system analysis and mathematical modelling of hierarchies [11, 15, 25, 31], the first and fourth rows are removed from Table 2, and the numbers 1 and 4 are deleted in the second and third columns. The resulting table is suitable for calculating the next iteration (Table 3).

Table 3. xxx

i	$D(w_i)$	$P(w_i)$	$D(w_i) \cap P(w_i)$
2	2	1,2,3,4,8	2
3	2,3,8	1,3,4	3
5	5,6,7,8,9	1,4,5	5
6	6,7	5,6	6
7	7,8	5,6,7	7
8	2,8	1,3,4,5,7,8,9	8
9	8,9	4,5,9,10	9
10	9,10	1,10	10

Performing the actions described above with this and the following tables, a multilevel graphical model of the priority influence of factors on the vaccination intensity is obtained (Fig. 4).

The synthesized model abstractly represents the subjective opinions of experts on the factors that affect the vaccination intensity. As it can be seen from the figure, the factors “level of education” and “respondent age” are of the greatest importance, depending on which a hierarchical “pyramid” of factors – semantic factors – is formed. The vaccination experience, in general, confirms this theoretically obtained initial conclusion.

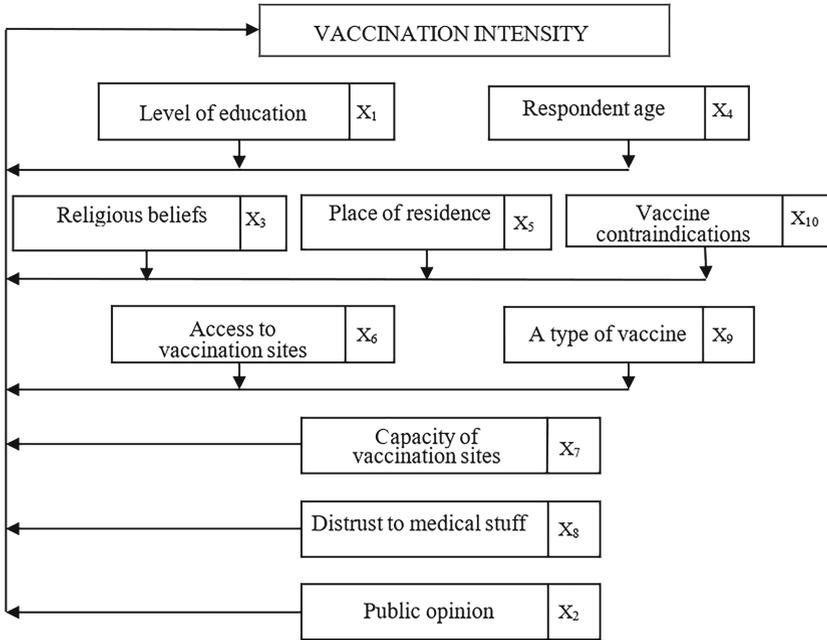


Fig. 4. A multilevel model of factors influencing the vaccination intensity

Since on the first three levels in Fig. 4 there are several factors that make it impossible to use the Pareto method to design alternatives, the method of pairwise comparisons [11, 25] is applied to the obtained model, the implementation of which will provide a strict ranking on the principle of “one level - one factor”. The result of pairwise comparisons will be a square inversely symmetric matrix, the processing of which according to a certain algorithm leads to the weight values of the factors. Finally, an optimized model of their influence on the vaccination process is synthesized as a prerequisite for determining the process intensity levels. The basis of the construction of this matrix is the scale of relative importance of objects [11, 25], shown in Table 4.

Taking into account the graphical model of Fig. 4 and the scale of relative importance of objects, a variant of the matrix of pairwise comparisons is obtained, presented in the form of a table for convenience (Table 5).

Table 4. Scale of relative importance of objects

Importance assessment	Comparison criteria	Explanation of the criteria selection
1	Objects are equivalent	Absence of preference k_1 over k_2
3	One object is dominated by another to some extent	There is a slight preference k_1 over k_2
5	One object prevails over another	There is a significant preference k_1 over k_2
7	One object is significantly dominated by another	There is a vivid preference k_1 over k_2
9	One object is absolutely dominated by another	Absolute preference k_1 over k_2 is doubtless
2,4,6,8	Compromise intermediates value	Auxiliary comparative assessments

Table 5. Matrix of pairwise comparisons

	X1 LE	X2 PO	X3 RB	X4 RA	X5 PR	X6 AV	X7 CV	X8 DM	X9 TV	X10 VC
X1, LE	1	9	3	2	3	5	6	8	5	3
X2, PO	1/9	1	1/6	1/9	1/6	1/5	1/4	1/3	1/6	1/8
X3, RB	1/3	6	1	1/3	1/2	3	5	6	3	2
X4, RA	1/2	9	3	1	1/3	5	8	9	6	3
X5, PR	1/3	6	2	3	1	1/5	6	8	3	2
X6, AV	1/5	5	1/3	1/5	1/5	1	3	6	2	1/3
X7, CV	1/6	4	1/5	1/8	1/6	1/3	1	3	1/4	1/6
X8, DM	1/8	3	1/6	1/9	1/8	1/6	1/3	1	1/8	1/9
X9, TV	1/5	6	1/3	1/6	1/3	1/2	4	8	1	1/3
X10, VC	1/3	8	1/2	1/3	1/2	3	6	9	3	1

The method proposed by Saati [11, 25] is used to obtain the main eigenvector of the matrix of pairwise comparisons, which will determine the numerical weights of the factors. The software implementation of the method leads to a normalized vector:

$$T_{norm} = (0, 264; 0, 014; 0, 117; 0, 196; 0, 134; 0, 058; 0, 028; 0, 016; 0, 056; 0, 112),$$

which essentially embodies the specified numerical priorities of the factors and establishes the preliminary formal result of the problem solution. For a visual integer representation of the weight values of the factors, the components of the

vector T_{norm} are multiplied by some scaling factor, for example $k = 1000$. The following representation of the factors weights is obtained:

$$T_{norm} = (264; 14; 117; 196; 134; 58; 28; 16; 56; 112).$$

The assessment of the obtained decision is determined by the consistency index:

$$IU = (\lambda_{max} - n)/(n - 1). \quad (10)$$

Calculations according to the software present $\lambda_{max} = 11,09$. According to (10) one gets: $IU = 0,12$. The values of the consistency index are compared with the reference values of the consistency index – a random index WI that depends on the number of objects being compared. In this case, the results are considered satisfactory if the index value does not exceed 10% of the reference value of the index for the corresponding number of objects. In our version, the reference value of the random index for ten factors is $WI = 1,49$.

Comparing the calculated value of the consistency index and the tabular reference value of the index for ten objects, and verifying the inequality $IU < 0,1 \times WI$, one obtains: $0,12 < 0,1 \times 1,49$. The inequality performance confirms the adequacy of the problem solution.

Additionally, the results are assessed by the consistency ratio, the value of which is obtained from the expression: $WU = IU/WI$. Because $WI = 1,49$, accordingly $WU = 0,08$. The results of pairwise comparisons can be considered satisfactory if $WU \leq 0,1$. Thus, the author have a sufficient level of process convergence and proper consistency of expert judgments represented in the semantic network and the matrix of pairwise comparisons.

Finally, a model of priority influence of factors on the vaccination process intensity is obtained, optimized according to the above criteria.

The obtained model becomes the basis for the experimental part, the essence of which is to design alternative options to the vaccination process and determine the optimal one.

5 Experiment

For further research according to the logic of the **method of linear convolution of criteria** from the generalized set (1) on the basis of the model in Fig. 5, the factors are singled out that correspond to the Pareto principle, the essence of which is to separate those factors from the general set that dominate over others in their influence, i.e. factors with lower weight priorities are excluded from further consideration. Its application simplifies the process of forming

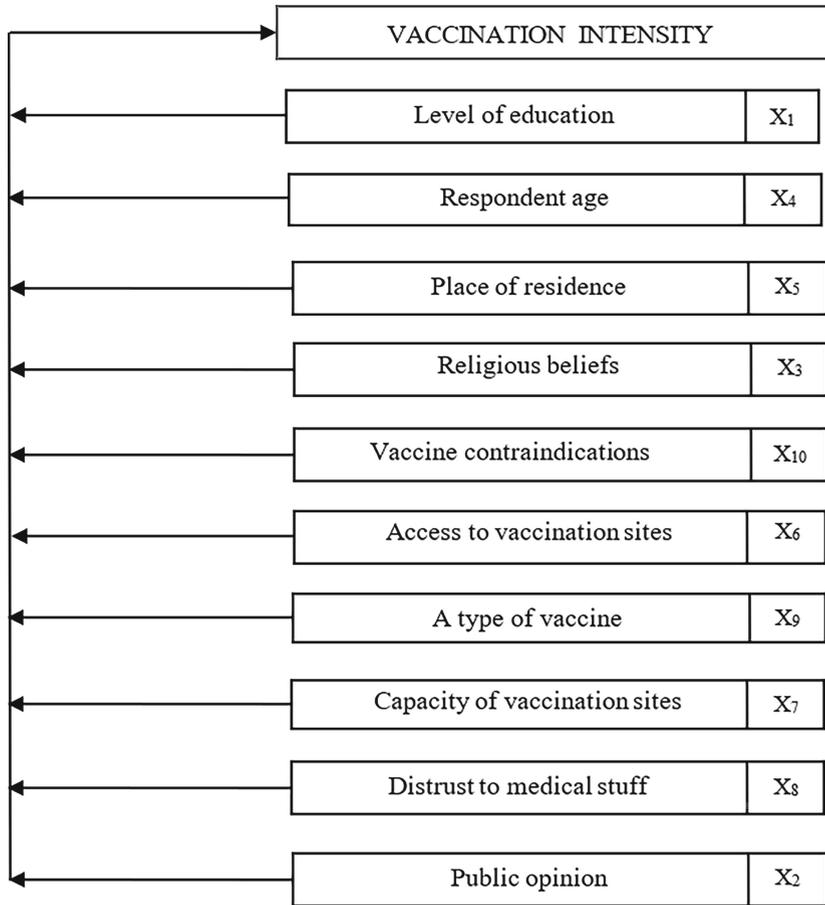


Fig. 5. An optimized model of priority influence of factors on the vaccination intensity

alternatives and provides sufficient quality of the result for practical use. Therefore, the following factors are selected for further research:

- x_1 is the level of education (LE, the factor weight is 264 c. u.);
- x_4 is the respondent age (RA, the factor weight is 196 c. u.);
- x_5 is the place of residence (PR, the factor weight is 134 c. u.).

Since the selected factors form a separate set, they are given new formal notations – F_1, F_2, F_3 .

A table of alternatives assessment is formed, keeping the names of factors, their weight values calculated previously and setting the combinations of factors effectiveness shares in options corresponding to the vaccination intensity levels: F_1 – low, F_2 – medium, F_3 – high. The shares of “participation” of factors in

the alternatives are set as a percentage scale with a step of 10 units. It is clear that there may be many such options, but for the experiment, one of them will be selected and displayed in the table of alternatives assessment.

Table 6. Alternatives assessment by Pareto set factors

Names of factors	Factor weight (c. u.)	Alternatives assessment by factors (in %)		
		r_1	r_2	r_3
Level of education F_1 (LE)	264	10	20	70
Respondent age F_2 (RA)	196	20	60	20
Place of residence F_3 (PR)	134	50	30	20

Table 6 presents the initial data for calculations: u_{ij} - is a utility of the j -th alternative $j = (1, 2, 3)$ according to the i -th factor $i = (1, \dots, 3)$; U_j - is a multifactor utility assessment of the j -th alternative. It is to remember that according to the expression (4) functional U_j gets this representation:

$$U_j = \sum_{i=1}^3 w_i u_{ij}; \quad j = 1, 2, 3. \tag{11}$$

Since the alternative factors are a new set, the weight values are calculated for them by constructing a matrix of pairwise comparisons based on their initial weights shown in Table 6.

As a result of processing of a matrix the specified values of factors weights are obtained $w_1 = 0, 6$; $w_2 = 0, 3$; $w_3 = 0, 1$; with the values of criteria: $\lambda_{max} = 3, 1$; $IU = 0, 08$; $WU = 0, 1$. The obtained values indicate the reliability of the results. New weight values of factors will be used in further procedures (Table 7).

Table 7. Matrix of pairwise comparisons for alternative factors

	LE	RA	PR
LE	1	4	5
RA	1/4	1	4
PR	1/5	1/4	1

On the basis of dependence (11), the expressions (12) are formed for the calculation of numerical values of multifactor utility assessments of alternatives U_j in options r_1, r_2, r_3 :

$$\begin{aligned}
 U_1 &= w_1 \cdot u_{11} + w_2 \cdot u_{21} + w_3 \cdot u_{31}; \\
 U_2 &= w_1 \cdot u_{12} + w_2 \cdot u_{22} + w_3 \cdot u_{32}; \\
 U_3 &= w_1 \cdot u_{13} + w_2 \cdot u_{23} + w_3 \cdot u_{33}.
 \end{aligned} \tag{12}$$

The maximum value of the utility functions $U_j(j = 1, 2, 3)$ of the combined partial target functionals determines the optimal vaccination option among the alternatives.

Confirmation of the adequacy of the theoretically substantiated option is performed using **the method of multi-criteria optimization**, in which the degree of pairwise preferences between alternatives is set by fuzzy logical relations. The essence of multi-criteria optimization will be to convolute many criteria (factors) into one scalar, which uses the method of intersection.

A summary of the algorithm for obtaining the optimal option is as follows [27]. For a set of alternatives $R\{r_1, r_2, r_3\}$, a fuzzy relations of preferences between the options $F_j(j = 1, 2, 3)$ are attached to each of the factors of the Pareto set which will correspond to the weight values of the factors w_j and the corresponding membership functions $\mu_j(x, y)$, calculated by the rule (13):

$$\mu_j(x, y) = \begin{cases} 1, & \text{if } f_j(x) \geq f_j(y), \text{ i.e. } (x, y) \in F_j \\ 0, & \text{if } (x, y) \notin F. \end{cases} \tag{13}$$

A convolution of fuzzy relations of preferences between options $Z_1 = \bigcap_{j=1}^3 F_j$ is found and a set of non-dominant alternatives Z_1^{nd} with membership functions are determined:

$$\mu_{Z_1}^{nd}(x) = 1 - \sup_{y \in X} \left\{ \sum_{j=1}^3 \mu_{Z_1}(y, x) - \mu_{Z_1}(x, y) \right\}. \tag{14}$$

Similarly, a set of non-dominant alternatives for convolution Z_2 is defined:

$$\mu_{Z_2}^{nd}(x) = 1 - \sup_{y \in X} \left\{ \sum_{j=1}^3 \mu_{Z_2}(y, x) - \mu_{Z_2}(x, y) \right\}. \tag{15}$$

Finally, the intersection of sets Z_1^{nd} and Z_2^{nd} is found, i.e. $Z_{nd} = Z_1^{nd} \cap Z_2^{nd}$, the maximum value of the membership function of which:

$$\mu_{nd}(x) = \min \{ \mu_{Z_1}^{nd}(x), \mu_{Z_2}^{nd}(x) \} \tag{16}$$

establishes an alternative that will determine the optimal COVID-19 vaccination.

6 Results and Discussion

The values of the utility functions u_{ij} of factors are calculated using **the method of linear convolution of criteria**. To do this, matrices of pairwise comparisons are constructed that will reflect the preferences of alternatives for each of the factors. The consistency of the results will be checked using the maximum eigenvalue of the priority vector λ_{max} for each of the matrices, the consistency index IU and the consistency ratio WU , the limit values of which are presented above. To form matrices on a scale of relative importance of objects, Table 6 is

used - data of the alternatives assessment for selected factors. The estimated values of alternatives will be compared according to the factors listed in the last three columns of the table.

As a result, the matrix of pairwise comparisons for the factor “Level of education” (LE), taking into account the percentage shares of its effectiveness in the alternatives will look like this:

LE	r_1	r_2	r_3
r_1	1	1/2	1/6
r_2	2	1	1/5
r_3	6	5	1

The calculation results are: $\lambda_{max} = 3,03$; $IU = 0,01$; $WU = 0,03$. The alternative utility by the factor “Level of education” is (LE): $u_{11} = 0,102$; $u_{12} = 0,172$; $u_{13} = 0,725$.

The factor “Respondent age” (RA) will cause this matrix:

RA	r_1	r_2	r_3
r_1	1	1/4	1
r_2	4	1	4
r_3	1	1/4	1

After its processing $\lambda_{max} = 3$; $IU = 0$; $WU = 0$. The alternative utility function by the factor “Respondent age” (RA) is: $u_{21} = 0,166$; $u_{22} = 0,666$; $u_{23} = 0,166$.

As to the factor “Place of residence” (PR) one gets:

PR	r_1	r_2	r_3
r_1	1	3	2
r_2	1/3	1	2
r_3	1/2	1/2	1

After its processing $\lambda_{max} = 3,14$; $IU = 0,07$; $WU = 0,1$. The alternative utility function by the factor “Place of residence” (PR) is: $u_{31} = 0,547$; $u_{32} = 0,263$; $u_{33} = 0,189$.

The criteria for assessing the reliability of calculations, namely the maximum values of priority vectors λ_{max} , the consistency indices IU and the consistency ratios WU meet the requirements. Substituting the weights and values of the

utility functions of factors in the formula (12), the final values of the combined functionals are obtained that characterize the priority of alternative COVID-19 vaccination levels:

$$\begin{aligned}
 U_1 &= 0,6 \cdot 0,102 + 0,3 \cdot 0,166 + 0,1 \cdot 0,547; \\
 U_2 &= 0,6 \cdot 0,172 + 0,3 \cdot 0,666 + 0,1 \cdot 0,263; \\
 U_3 &= 0,6 \cdot 0,725 + 0,3 \cdot 0,166 + 0,1 \cdot 0,189.
 \end{aligned}
 \tag{17}$$

Finally, one gets: $U_1 = 0,166$; $U_2 = 0,329$; $U_3 = 0,504$. As a result, among the developed alternatives for determining the vaccination intensity, the third option will be considered optimal taking into account the maximum value of the combined functional U_3 . Theoretically, such an option should ensure the proper level of the vaccination process, i.e. to form an important initial part of its quality.

Confirmation (or denial) of the obtained result is performed using the **method of multi-criteria optimization**. Taking into account the algorithm for selecting an alternative by this method, the best option for determining the vaccination level is defined. A set of alternatives $R = \{r_1, r_2, r_3\}$ is established. Non-dominant factors of the Pareto set are used, which determine the set of preferences relations, and the corresponding weights of factors for the convolution Z_2 , namely: $w_1 = 0,6$; $w_2 = 0,3$; $w_3 = 0,1$.

The preference relation in options is established by each factor.

Level of education: (F_1): $r_1 < r_2, r_1 < r_3, r_2 < r_3$.

Respondent age: (F_2): $r_1 < r_2, r_1 = r_3, r_2 > r_3$.

Place of residence: (F_3): $r_1 > r_2, r_1 > r_3, r_2 > r_3$.

Since the calculation of the optimal vaccination option by this method is performed by a special application, the attention is paid only to the key points.

Thus, according to the presented relations of preferences, matrices of relations are constructed, on the basis of which the convolution of relations Z_1 and the additive convolution of relations Z_2 is determined. By the expression (14), for each of the alternatives, the following values of membership functions are obtained:

$$\begin{aligned}
 \mu_{Z_1}^{nd}(r_1) &= 1 - \sup_{y \in R} \{0 - 0; 0 - 0\} = 1; \\
 \mu_{Z_1}^{nd}(r_2) &= 1 - \sup_{y \in R} \{0 - 0; 0 - 0\} = 1; \\
 \mu_{Z_3}^{nd}(r_1) &= 1 - \sup_{y \in R} \{0 - 0; 0 - 0\} = 1.
 \end{aligned}$$

Taking into account the calculation results, one gets: $\mu_{Z_1}^{nd}(r_i) = [1; 1; 1]$.

Similarly, a set of non-dominant alternatives is defined for the relation Z_2 according to (15):

$$\begin{aligned} \mu_{Z_2}^{nd}(r_1) &= 1 - \sup \{(0, 3 - 0, 7); (0, 9 - 0, 1)\} = 0, 2; \\ \mu_{Z_2}^{nd}(r_2) &= 1 - \sup \{(0, 9 - 0, 7); (0, 3 - 0, 7)\} = 0, 8; \\ \mu_{Z_2}^{nd}(r_3) &= 1 - \sup \{(0, 4 - 0, 9); (0, 4 - 0, 9)\} = 1. \end{aligned}$$

As a result, the membership function $\mu_{Z_2}^{nd}(r_i) = [0, 2; 0, 8; 1]$ is obtained.

The last step is to find the convolution of the intersection of the sets Z_1^{nd} and Z_2^{nd} , i.e. $Z_{nd} = Z_1^{nd} \cap Z_2^{nd}$. Taking into account the fact that $\mu_{Z_1}^{nd}(r_i) = [1; 1; 1]$ and $\mu_{Z_2}^{nd}(r_i) = [0, 2; 0, 8; 1]$ the final membership function $\mu_Z^{nd}(r_i) = [0, 2; 0, 8; 1]$ is obtained, the maximum value of which determines the optimal option.

The convolution membership function Z indicates that the optimal alternative to the vaccination process with the above-mentioned preference relation of the factors utility is the option r_3 , whose membership function has the maximum value. The obtained conclusion confirms the reliability of the study, as the optimal option of the vaccination intensity, calculated above by the method of linear convolution of criteria, is also the third option (Figs. 6, 7 and 8).

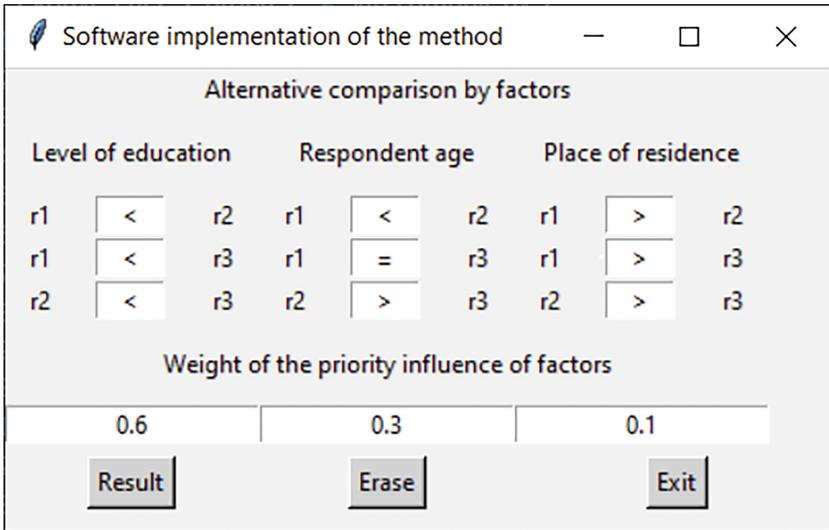


Fig. 6. The window of initial preference relations of alternatives

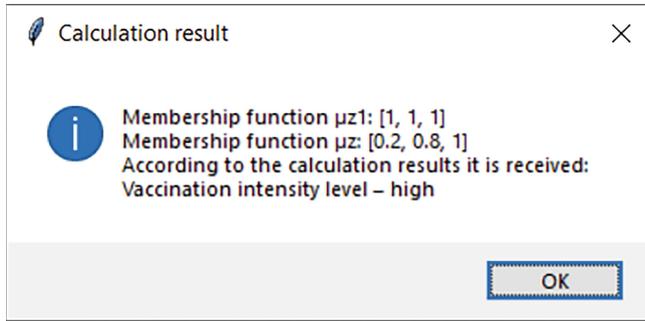


Fig. 7. The calculation result of the vaccination intensity levels

Comparison of vaccination intensity levels

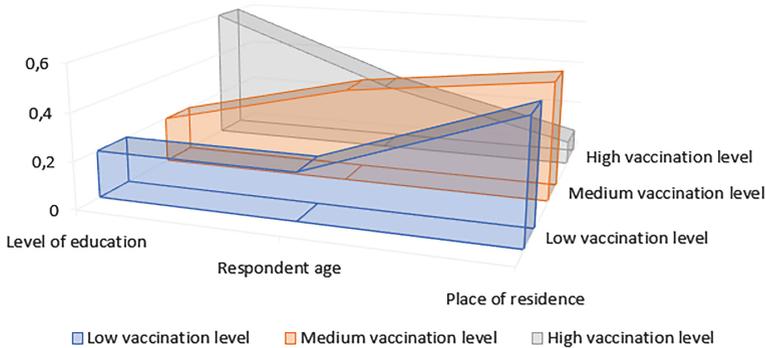


Fig. 8. Histogram comparing the vaccination intensity levels

7 Conclusions

A review of literature sources is done related to the problem of COVID-19 origin and spread, which in addition to the medical direction violates the socio-economic component that affects all spheres of the society activity. There is a lack of scientific research on this issue, which necessitates the formation and study of factors relevant to the process of COVID-19 vaccination, as one of the most effective means of combating the pandemic. The study has revealed the essence of analytical and algorithmic software suitable for the formation of alternatives to the vaccination process and determine the levels of its intensity in the presence of a set of factors, sorted by the importance of their impact on the process. Methods of operations research theory have been used for theoretical grounding and the effective design of alternatives.

A formalized representation of the relationships between factors has been made using the semantic network as an information database, which has become a prerequisite for ranking factors by their weight criteria. Using the methodol-

ogy of hierarchies modelling, the levels of factors preferences are established and a multilevel model of their priority influence on the researched process is synthesized. According to the method of linear convolution of criteria and the method of multi-criteria optimization, alternative options of the vaccination process are designed and the prognostic assessment of the levels of COVID-19 vaccination intensity is performed, which allows the program calculation of the optimal option for the specific initial parameters.

The obtained results determine the possibility of their practical application in planning the vaccination process in order to obtain the predicted level of its intensity, taking into account the personal characteristics of the vaccinated contingent, the residence region, medical prerequisites.

The prospect of further development of this direction could be to create an automated system for forming the level of COVID-19 vaccination intensity on the basis of the received and processed statistics within Ukraine, which would take into account a wider range of factors influencing the process effectiveness.

References

1. Ahmad, W., Abbas, M., Rafiq, M., Baleanu, D.: Mathematical analysis for the effect of voluntary vaccination on the propagation of corona virus pandemic. *Results Phys.* **31**, 104917 (2021)
2. Al-Amer, R., et al.: Covid-19 vaccination intention in the first year of the pandemic: a systematic review. *J. Clin. Nurs.* **31**(1–2), 62–86 (2022)
3. Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M.: A survey of ontology learning techniques and applications. *Database* **2018** (2018)
4. Bayes, C., Valdivieso, L., et al.: Modelling death rates due to covid-19: A bayesian approach. arXiv preprint [arXiv:2004.02386](https://arxiv.org/abs/2004.02386) (2020)
5. Below, D., Mairanowski, F.: The impact of vaccination on the spread patterns of the covid epidemic. *medRxiv* (2021)
6. Bloom, D.E., Canning, D., Weston, M.: The value of vaccination. In: *Fighting the Diseases of Poverty*, pp. 214–238. Routledge (2017)
7. Buonomo, B.: Effects of information-dependent vaccination behavior on coronavirus outbreak: insights from a siri model. *Ricerche mat.* **69**(2), 483–499 (2020)
8. Campos-Mercade, P., Meier, A.N., Schneider, F.H., Meier, S., Pope, D., Wengström, E.: Monetary incentives increase covid-19 vaccinations. *Science* **374**(6569), 879–882 (2021)
9. Chimmula, V.K.R., Zhang, L.: Time series forecasting of covid-19 transmission in Canada using LSTM networks. *Chaos, Solitons Fractals* **135**, 109864 (2020)
10. Cihan, P.: Forecasting fully vaccinated people against covid-19 and examining future vaccination rate for herd immunity in the US, Asia, Europe, Africa, South America, and the world. *Appl. Soft Comput.* **111**, 107708 (2021)
11. Hilorme, T., Tkach, K., Dorenskyi, O., Katerna, O., Durmanov, A.: Decision making model of introducing energy-saving technologies based on the analytic hierarchy process. *Journal of Management Information and Decision Sciences* **22**(4), 489–494 (2019)
12. Khadir, A.C., Aliane, H., Guessoum, A.: Ontology learning: Grand tour and challenges. *Comput. Sci. Rev.* **39**, 100339 (2021)

13. Kyrychko, Y.N., Blyuss, K.B., Brovchenko, I.: Mathematical modelling of the dynamics and containment of covid-19 in Ukraine. *Sci. Rep.* **10**(1), 1–11 (2020)
14. Marchau, V.A., Walker, W.E., Bloemen, P.J., Popper, S.W.: *Decision making under deep uncertainty: from theory to practice.* Springer Nature (2019)
15. Marden, J.R., Shamma, J.S.: Game theory and control. *Ann. Rev. Control Robot. Autonomous Syst.* **1**, 105–134 (2018)
16. Nesteruk, I.: Visible and real sizes of the covid-19 pandemic in ukraine (2021)
17. Paul, A., Sikdar, D., Mahanta, J., Ghosh, S., Javed, M.A., Paul, S., Yeasmin, F., Sikdar, S., Chowdhury, B., Nath, T.K.: Peoples' understanding, acceptance, and perceived challenges of vaccination against covid-19: a cross-sectional study in Bangladesh. *PLoS ONE* **16**(8), e0256493 (2021)
18. Pavlyuk, O., Fedevich, O., Strontsitska, A.: Forecasting the number of patients with covid-19 in LVIV region. *Bull. Vinnytsia Polytech. Inst.* **3**, 57–64 (2020)
19. Peng, L., Guo, Y., Hu, D.: Information framing effect on public's intention to receive the covid-19 vaccination in china. *Vaccines* **9**(9), 995 (2021)
20. Perumal, V., Narayanan, V., Rajasekar, S.J.S.: Detection of covid-19 using CXR and CT images using transfer learning and Haralick features. *Appl. Intell.* **51**(1), 341–358 (2021)
21. Sarica, S., Luo, J.: Design knowledge representation with technology semantic network. *Proc. Design Soc.* **1**, 1043–1052 (2021)
22. Semerikov, S., et al.: Our sustainable coronavirus future (2020)
23. Senkivskyy, V., Kudriashova, A.: Multifactorial selection of alternative options for an edition design based on a fuzzy preference relation. *Printing and Publishing* **73**(1), 80–86 (2017)
24. Senkivskyy, V., Pikh, I., Kudriashova, A., Lytovchenko, N.: Theoretical fundamentals of quality assurance of publishing and printing processes (part 2: Synthesis of priority models of factors action). *Printing and Publishing* **71**(1), 20–29 (2016)
25. Senkivskyy, V., Pikh, I., Babichev, S., Kudriashova, A., Senkivska, N.: Modeling of alternatives and defining the best options for websites design. In: *IntellITSIS*, pp. 259–270 (2021)
26. Shekera, O.: Analytical review of the global coronavirus infection pandemic in Ukraine. *Health Soc.* **10**(1), 14–25 (2021)
27. Taha, H.A.: *Operations Research: An Introduction*, vol. 790. Pearson/Prentice Hall Upper Saddle River, NJ (2011)
28. Troiano, G., Nardi, A.: Vaccine hesitancy in the era of covid-19. *Public Health* **194**, 245–251 (2021)
29. Walkowiak, M.P., Walkowiak, D.: Predictors of covid-19 vaccination campaign success: lessons learnt from the pandemic so far. a case study from poland. *Vaccines* **9**(10), 1153 (2021)
30. Yang, L., Cormican, K., Yu, M.: Ontology-based systems engineering: a state-of-the-art review. *Comput. Ind.* **111**, 148–171 (2019)
31. Zhu, Q., Rass, S.: Game theory meets network security: a tutorial. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2163–2165 (2018)



Application of the Theory of Functional Stability in the Problems of Covering Territories by Sensory Networks

Oleh Mashkov¹, Alexey Bychkov², Ganna Kalahnik³,
Victor Shevchenko², and Svitlana Vyshemyrska⁴^(✉)

¹ State Ecological Academy of Postgraduate Education and Management, Kyiv, Ukraine

² Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

³ Flying Academy of the National Aviation University, Kropyvnytskyi, Ukraine

⁴ Kherson National Technical University, Kherson, Ukraine

vyshemyrska.svitlana@kntu.net.ua

Abstract. The survivability of sensor networks as a component of functional stability in the tasks of covering the territory is considered in the article. It is proposed to consider survivability as a component of functional stability of networks. The survivability of the sensor network is presented as the property of the network to provide coverage and data transmission in case of loss or failure of network components such as sensors and connections between sensors. Ensuring survivability is achieved by adding additional sensors to the network and connections between sensors. The approach of optimization of energy consumption of the network with additional sensors by adjusting the radii of coverage of additional sensors depending on the operating condition of the main sensors is presented. The results of computational experiments confirm the effectiveness of the proposed approaches.

Keywords: Sensors · Sensor networks · Functional stability · Survivability · Safety margin · Functional stability limit

1 Introduction

Quite a large number of publications in the field of functionally stable systems determine the need to analyze the results and their interpretation in the general context of cybernetics. The authors propose a description of this scientific field, a description of the main results and prospects for further development of the theory of functional stability of dynamic systems.

The problematic situation that led to the emergence of this scientific area is associated with the development in the late twentieth century of complex autonomous technical systems operating in extreme conditions (primarily aerospace and rocket and space systems). Their high cost and potential danger required an appropriate level of reliability and safety.

The experience of creating and using information systems in recent decades has revealed many situations in which damage and failure of their operation were due to defects in software packages and were accompanied by great damage. As a result of errors in automatic control programs, several domestic, American and French satellites were killed, and failures and catastrophes occurred in complex administrative, banking and technological information systems.

As a result, about twenty years ago, the first generalizing works appeared, in which the concept and basic provisions of the theory of software reliability for information systems were formulated. At this time, the foundations of methodology and technology for creating reliable software packages were laid, many methods and models for studying the reliability of complex systems were created, but a single approach to solving this problem was not provided. The reason for this is the uniqueness of each software system. However, when developing responsible projects, their developers and customers try to obtain an assessment of the reliability of the software, usually based on the results of final tests.

At the same time, traditional methods based on multiple redundancy, the introduction of built-in control systems and elements with a high level of reliability, deteriorated the technical and economic characteristics of the designed systems, without leading to the necessary reduction of hazardous situations. The need to introduce additional hardware redundancy to ensure system reliability has become an important limitation of this approach.

It is proposed to consider the extraordinary states of a complex system, which are caused by failures, as acceptable, and for them to form adequate (functionally stable) management aimed at eliminating the consequences of failures and maintaining the functions of the system. This management ensures the redistribution of system resources to achieve the main goal, even in the event of failure.

An important condition for ensuring this quality is the ability to redistribute available resources of different physical nature within the system. It was found that ensuring the functional stability of systems goes beyond the traditional for the classical territory of automatic control optimization tasks “in small” (on a given software trajectory control is determined by the optimization of transients according to certain criteria), as it provides incomplete a priori information about object, operative formation of a program trajectory for a current situation and optimum use of all available resources.

In this research, the authors propose to consider functional stability as a property of a complex dynamic system, which consists in the ability to perform at least a set minimum of its functions for failures in the information computing, control, energy part of the system and external influences not provided by operating conditions.

Therefore, the task of ensuring functional stability can be considered as a kind of tasks of adaptive optimal management “in large”, providing optimal use at each stage or mode of operation of the system of all available resources (energy, information, computing), to achieve the main goal for this stage restrictions.

2 Related Works

In technical cybernetics, namely in the theory of automatic control, the classical theory of stability of dynamic systems is built. In this theory, we can estimate the stability without solving a system of differential equations describing the object, but using simple features, conditions and criteria of stability [14,20]. By analogy with the classical theory of stability, it is proposed to estimate the functional stability by the parameters of the graph describing the structure of the data exchange system. It turns out that the appearance of the graph and its parameters can determine whether the system will be functionally stable, unstable or neutral.

Many authors have been involved in the development and formation of the conceptual apparatus and applied approaches to functional stability [2,3]. The peculiarity is the formation of this device on the examples of ensuring the functional stability of specific types of technical systems. The basic concepts of the theory of functional stability are given in [13–15]. The introduced concepts were used to ensure the functional stability of aerospace systems and aircraft. The issue of ensuring the functional stability of intelligent flight control systems of aircraft is considered in [15,22,23]. The concept of functional stability of technical human-machine systems is considered in [12,13]. Ensuring functional stability using artificial intelligence and redundancy is considered in [11,19]. It should also be noted the possibility of applying the theory of functional stability and ensuring functional stability in various technological fields: metallurgy and computer networks, distributed information systems [4,5,7,8,24]. Functional stability due to the excess of information resources in the nodes of information systems in the context of computer epidemics is presented in [1,26]. There is a difference between the concepts of stability of functioning and functional stability [14,15].

The approaches considered in the works of most authors can be used to ensure the functional stability and survivability of sensor networks. However, all the authors analyzed did not consider specific applied approaches to ensure the survivability and functional stability of sensor networks.

3 Material and Methods

The article considers the survivability of sensor networks based on the theory of functional stability. The peculiarity of the approach under consideration lies in the fact that the object of research is the possibility of creating functionally stable information and telecommunication systems that can identify and parry the factors of violations of their functioning.

The subject of research is the process of functioning of sensory systems in the event of emergency and emergency situations.

It is proposed to decompose the methods of construction of functionally stable complex dynamic systems into methods of detection and methods of parrying failures in information and telecommunication systems.

The methods of the first group allow to a priori form images-standards of consequences of abnormal situations caused by failures of functional subsystems of the complex and to store them in memory.

The methods of the second group allow (after determining the image of the current state and comparing it with the stored images-standards) to draw a conclusion about the redistribution of resources of the complex to ensure the proper functioning of the components of the complex in the presence of a failure.

According to each case, adequate restorative management is formed.

The special significance of this approach is that it allowed to divide the general task of synthesis of functionally stable systems into separate tasks: failure detection and failure parrying.

Problems of ensuring functional stability are also important for information systems, where the failure of at least one of the elements leads to the loss of the ability to perform operational tasks. Simple introduction of redundancy (duplication of elements) leads to more expensive system, but does not guarantee sufficient stability of a complex system.

3.1 Criteria and Indicators of Functional Stability of a Complex System

The conducted research allowed to establish the connection between the concept of “functional stability” and the concepts of “reliability”, “survivability” and “fault tolerance”. The fundamental difference between them is shown: methods of ensuring functional stability are aimed not at reducing the number of failures and violations (as traditional methods of increasing the reliability, survivability and fault tolerance of technical systems), but at ensuring the most important functions when these violations have already occurred.

The peculiarity of the methods of synthesis of functionally stable on-board information and control complexes is that they do not consider the processes that led to failures. For the formation of special management is important the very fact of disruption of some element of the complex. Thus, for a complex system described by nonlinear equations and failures in the form of abrupt changes in structure, a graph criterion of functional stability is proposed.

The functioning of information and telecommunication systems takes place in constant interaction with the external environment. At the same time, a large class of such interaction represents a variety of conflicts that significantly affect the achievement of the system-wide goal. Such (information) conflicts lead to the destruction of information resources, disruption of regular information processes, and as a consequence of disruption of system and application functions.

All this determines the presence in information systems of mechanisms that should provide a new quality - the ability to preserve and/or restore data (stability) of functions under various adverse effects.

This quality is called the functional stability of information systems. Functional stability can be considered as an integral property that includes reliability, survivability and safety. Accordingly, the assessment of FS indicators required to compare different design options is a complex scientific and practical task.

It is even more difficult to find the best option, which achieves optimal FS performance with some restrictions. A number of scientific and practical results have been obtained in this direction, but the lack of educational and methodological literature on this important issue leads to certain shortcomings in the methodology of developing promising complex systems taking into account possible abnormal, emergency and catastrophic situations.

Modern software systems use a huge amount of data that passes through standard modules and functions. Therefore, to find all the connections and ways of processing information, even for a fairly simple program is almost impossible. Based on this, the detail of the elements of reliability calculation is usually limited to completed software formations (software modules) that interact with each other and form a complex system.

Experimental determination of the real reliability of software operation is very time consuming, difficult to automate and not always safe for the life cycle of a complex system.

At present, considerable experience has been gained in determining the reliability of complex systems used in aviation, space and other fields of modern highly intelligent technology. In these areas, it is unacceptable to use the operation of real objects to test and determine the reliability of complex systems.

As a result, methods and tools for modeling the environment for automated test generation in testing the reliability of such complex systems have become particularly important. In these cases, on the basis of software models and components of real systems, modeling test benches are created, which provide an opportunity to determine the reliability of specific complex systems in normal and critical external influences that meet the true characteristics of the environment.

Let's define reliability of the software of information system as possibilities of a complex of programs to carry out the set functions, keeping in time values of the established indicators within the set limits. During operation, the change in the reliability of the software over time differs significantly from the change in the reliability of hardware.

The software is not prone to wear, there are virtually no manufacturing errors, as they are rare and can be easily corrected. Software unreliability is entirely determined by design errors.

In this case, functional stability means the property of the system to perform its functions for a specified time under the influence of the flow of operational failures, intentional damage, interference in the exchange and processing of information, as well as errors of staff [15, 22, 23].

In fact, the functional stability of a complex technical system combines the properties of reliability, fault tolerance, survivability and characterizes the ability of the object to restore serviceability through the use of redundancy.

The mathematical model for representing the structure of a complex system has the form of an undirected graph $G(V, E)$, $v_i \in V$, $e_{ij} \in E$, $i, j = 1, \dots, n$, described by the adjacency matrix. The set of vertices V corresponds to the set of

switching nodes of dimension n , and the set of edges E - the set of communication lines between the switching nodes.

It is assumed that a complex system will perform the main function - data exchange, if between any pair of switching nodes there is at least one route of information transfer. Thus, the graph connectivity requirement provides a basis for quantifying the functional stability property of a data exchange system.

However, this paper does not consider the quality of the main functions, described by the delay time of the message when sending. It is also assumed that the communication channels have a bandwidth that allows you to send any amount of information flow.

A Sign of Functional Stability of the Structure. A structure of a complex system is functionally stable if the graph of the structure is one-component and has no bridges and junctions. The inverse definition allows to determine the functional instability of the structure.

A Sign of Functional instability of the Structure. The structure of a complex system is functionally unstable if its graph is multicomponent and unconnected.

Thus, the appearance of the graph, namely the number of components, the presence of bridges and junctions of the graph, can be judged on the functional stability of the structure, i.e. the inherent ability to fend off failures and damage. However, for highly branched and multi-vertex graphs, it is difficult to evaluate the appearance. Therefore, to quantify the degree of functional stability, we will consider the indicators of *functional stability of the structure*:

1. The number of vertex connectivity $\chi(G)$ is the smallest number of vertices, the removal of which, together with the incident edges, leads to an incoherent or single-vertex graph.
2. The number of edge connections $\lambda(G)$ is the smallest number of edges, the removal of which leads to an incoherent graph.
3. Connectivity probability $P_{ij}(t)$ is the probability that a message from node i to node j will be transmitted in no more than t .

The analysis of these indicators allows us to highlight the following features:

- vertex and edge connectivity numbers characterize only the current structure, regardless of the reliability of switching nodes or communication lines;
- indicators $\chi(G)$ and $\lambda(G)$ take the values of integers and related relationships

$$\chi(G) \leq \lambda(G);$$

- the probability of connectivity $P_{ij}(t)$ allows you to take into account the reliability of switching equipment, the type of physical information transmission channel, the availability of backup channels and routes, as well as the connectivity of the distributed structure. However, calculating the value $P_{ij}(t)$ is a complex and cumbersome task;

- the probability of connectivity characterizes only the connectivity between one pair of vertices. In order to characterize the connectivity between all pairs of vertices, it is necessary to operate with the adjacency matrix:

$$A = \|a_{ij}\|, \quad i, j = 1 \dots n, \quad a_{ij} = \begin{cases} 1, & \text{for } e_{ij} \in E; \\ 0, & \text{for } e_{ij} \notin E. \end{cases}$$

On the basis of the offered signs and indicators it is possible to develop criteria of functional stability of structure:

1. The structure will be functionally stable if the number of vertex connectivity satisfies the condition:

$$\chi(G) \geq 2$$

2. The structure will be functionally stable if the number of edge connections satisfies the condition:

$$\lambda(G) \geq 2$$

3. The structure will be functionally stable if the probability of connectivity between each pair of vertices is at least given:

$$P_{ij}(t) \geq P_{ij}^{zad}, \quad i \neq j, \quad i, j = 1 \dots n,$$

where n is the number of vertices of the graph $G(V, E)$.

These criteria allow to determine the functional stability of the current structure of the data exchange system on the basis of accurate calculations.

At the boundary between the two regions of stability and instability, there is a specific region in which the system is not functionally stable and, at the same time, is not functionally unstable. This area, by analogy with the theory of stability of dynamic systems, will be called the limit of functional stability of the structure [14].

The following position is a sign of the *limit of functional stability*. The current structure is at the limit of functional stability, if the graph of the structure is connected, includes bridges ($N_E > 0$) or connection nodes ($N_V > 0$):

$$\{K = 1\} \wedge [\{N_V > 0\} \vee \{N_E > 0\}],$$

where: K is the number of components of the graph, and the condition $K = 1$ means that the graph is connected; $N_V(N_E)$ is the number of connection nodes (bridges) of the graph.

A bridge is an edge of a connected graph that connects two subgraphs, after the removal of which the graph is transformed from one-component to two-component. In some works on the theory of urban graphs is called the isthmus.

A connection node is a vertex of a connected graph, after the removal of which, together with its incident edges, the graph changes from one-component to two-component.

The presence in the structure of a bridge or junction connecting two subgraphs means that all routes of information transfer from the vertices of one

subgraph to the vertices of another will contain this bridge or junction. This event significantly reduces the structural reliability and functional stability of the data exchange system. Therefore, in order to bring the system into a functionally stable state, it is necessary to introduce backup communication lines in the structure so that there are no bridges or junctions in the structure. At the same time, several independent and alternative routes of information transfer will appear.

The analysis of structures shows that if the system is on the verge of stability, it is operational and performs the required range of functions. However, if at least one bridge or junction fails, the system becomes unstable.

Areas of functional stability and instability can also be represented in Cartesian space in the coordinates of N_E , N_V (Fig. 1). Depending on the parameters N_E , N_V of the graph of the structure is determined by a point on the plane that will characterize the state of the system. According to the affiliation of a point to a particular area, one can judge the functional stability or instability of the system. In the graphical representation, the limit of the functional stability of the system will be the geometric location of the points lying on the two lines $N_E = 1$ and $N_V = 1$.

Based on the introduced concepts, the question arises as to how far the current structure extends from the limit of stability or, on the other hand, what is the margin of functional stability. It can also be defined in terms of the coherence of the structure. In this regard, the stock will be characterized by the number of failures (rupture of edges or failure of vertices), which can lead the structure to an unstable state.

The margin of functional stability can be quantified on the basis of the following indicators:

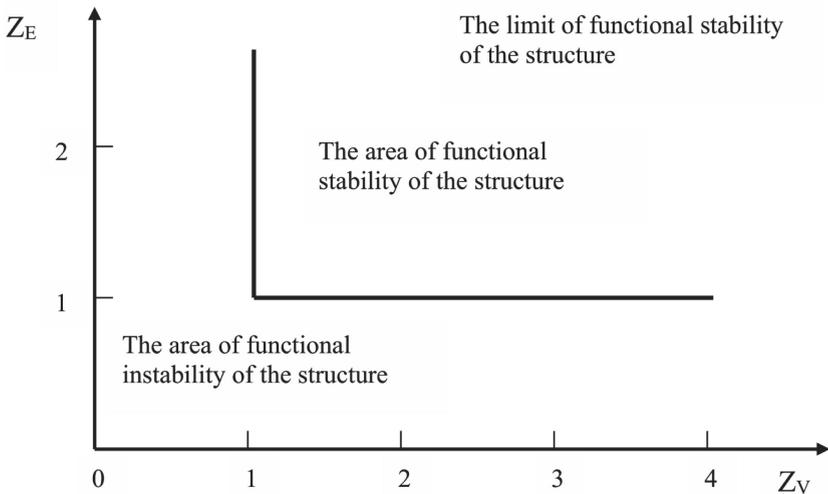


Fig. 1. Geometric interpretation of areas of functional stability and instability

1. Costal margin - the number of Z_E equal to the power of the minimum section, which translates the graph from one-component to two-component.
2. Vertex margin of stability - the minimum number of vertices of the Z_V graph, after removing which the graph changes from one-component to two-component.

The geometric interpretation of the stability margin will be defined as the minimum distance from a point on the plane defined by the parameters N_E , N_V , to the stability limit (Fig. 1).

You can also calculate the margin of functional stability by the probability of connectivity, as the difference between the set value and the current. Obviously, in this case, the stock will be expressed by a square matrix in which each element will have a value $(P_{ij}^{zad} - P_{ij})$.

Thus, on the basis of the signs of functional stability after determining the proposed parameters, it is possible to determine the state of the data exchange system, namely the finding of the system in a functionally stable state or functionally unstable. The degree of functional stability determines the margin of functional stability, which can be found both analytically by the proposed formulas and graphically (see Fig. 1). On the basis of research data there is a possibility: substantiation of requirements to data exchange systems to be designed, solving the problem of synthesis of optimal structure by the criterion of maximum functional stability with limitation on the cost of construction and operation of communication lines, and reasonable increase of data exchange structure operation.

The advantages of this approach are that you can quantify the functional stability of a complex system based on simple external features. Based on these assessments, it is possible to make recommendations for building the structure or to make reasonable requirements for the structure of the data exchange system to be designed.

3.2 Sustainability of Information and Telecommunication Systems in Emergency Situations

When designing information and telecommunication systems, it is necessary to provide for the operation of facilities in normal (regular), but also in extreme conditions caused by emergencies.

Extreme situations for the functioning of information and telecommunication systems can be created due to sudden changes in temperature, overpressure, electromagnetic and ionizing radiation, harmful pollution. These changes can lead to various deformations, damage, destruction, changes in the ecological balance of the environment, negative emotional phenomena, epidemics, disability, loss of life and other catastrophic phenomena (catastrophes and catastrophic consequences include natural disasters, major accidents, military conflicts in which there is a danger to human life).

Currently, the design of facilities and devices of information and telecommunication systems should consider the operation not only in normal but also in

extreme conditions, so in case of various emergencies should develop practical recommendations for restoring complex facilities and their elements.

Information and telecommunication systems occupy the most important place in the country's economy, because their normal operation ensures the management of economic and other activities in all conditions, and therefore one of the most important tasks is to ensure their sustainable operation in emergencies.

Under the stability of information and telecommunications systems we understand their ability to work in emergency, i.e. emergencies, and in case of violations of their work - is the ability to restore efficiency in the shortest possible time.

The concept of stability of information and telecommunication systems essentially includes two concepts: physical (static) and operational stability.

The physical stability of information and telecommunications systems or their elements means the physical strength of its elements, which may occur in emergencies.

Operational stability of information and telecommunication systems is understood as ensuring sustainable management of economic and other activities in case of emergencies, and in case of disruption - is the ability to resume work in the shortest possible time.

3.3 Technique to Ensure the Functional Stability of the Coverage of Territories by Sensor Networks

According to the latest data and research results, sensors and sensor networks have become widespread in the modern technological world [9,25]. Among the advantages of using sensors are small size, low power consumption, ease of operation and development of appropriate software. Sensors are used for measurement, monitoring, observation, accumulation and processing of data of various natures, etc. The main task described above is the monitoring task [15]. A key aspect of this task in the case of static sensors is the construction of a sensor network, the coverage area of which covers the required area. Also an important parameter is the correct location of sensors [22,23]. In turn, when using dynamic sensors, a key aspect is to build a trajectory that provides maximum coverage.

Simple fault tolerance is not enough to ensure the smooth operation of the sensor network. Elements of the information system may fail, but must ensure the performance of functions assigned to the system by reconfiguring system resources. That is, the sensor network must be functionally stable.

According to [14,15] functional stability is the ability of a system to perform its functions during a given time interval under the influence of operational failures, intentional damage, interference in the exchange and processing of information and errors of personnel. Functional stability of a complex technical system combines the properties of reliability, fault tolerance, survivability. The concept of functionality is a close concept to be considered, and reflects the set of functions performed by the system.

Functionality can be defined as the total useful effect of available system functions or as the total number of system functions, or as a fraction of functions that the system has retained at the moment, despite the harmful effects.

Functional instability is the inability of a system to meet the conditions of functional stability formulated above. We will consider a dysfunctional state as an identical concept. The causes of dysfunctional states can be external and internal, malicious and unintentional. Functional stability combines the properties of reliability, fault tolerance and survivability [22].

Reliability [21] is the property of the system to keep in time within the established limits the values of all parameters that characterize the ability to perform the desired functions in all specified modes and conditions.

Fault tolerance [21] is the property of the system to maintain performance in case of failure of its components.

Survivability [21] is the property of the system to maintain limited performance in the face of external influences that lead to failures of its components. Survivability characterizes the property of the system to resist the development of critical failures under any operating conditions, including those not provided for in the documentation.

The characteristics of the main properties of functional stability are presented in the following table (Table 1).

Table 1. Characteristics of the main properties of functional stability

Property	Reliability	Fault tolerance	Vitality
Content properties	The ability not to fail	Ability to work after failure of some elements	Ability to work in conditions of external harmful (malicious) influences
What you need to provide	Reliable components	Reservation	Redundancy and redistribution of functions
What you need to provide in case of an emergency	Reliable components, licensed software	Stock of information resources	Stock of information resources

Important concepts in the theory of functional stability are the concept of the limit and margin of functional stability. The limit of functional stability is the limit in the space of phase coordinates, which separates the states of functional stability and functional instability of the system. The margin of functional stability is the distance (a measure of distance) of the most probable states of the system from the limit of functional stability. The margin of functional stability shows how far the system is from the limit of functional stability, at the intersection of which the system moves from a functionally stable state to a functionally unstable state or vice versa.

Let's consider the sensor network S used to cover and monitor an area of $T m^2$. Elements of this network will have the following characteristics:

$$s_i = s_i(x_i, y_i, r_i, r_{max}),$$

where x_i, y_i are the coordinates of the sensor, r_i is the current coverage radius, r_{max} is the maximum coverage radius, $i = \overline{1, n}$, n is the number of sensors.

The survivability of the sensor network will be understood as the property of the network to provide coverage of the territory and data transmission in case of loss or failure of network components (sensors and connections between sensors).

Let's consider the case of ensuring survivability during the monitoring task. Denote by S_n the area covered by the sensor network consisting of n sensors. Obviously, each of the n sensors that provide coverage S_n has a set of characteristics from view (1). Let's disconnect a certain sensor from the network. In this case, the coverage area of the network will be equal

$$S_{n-1} < S_n.$$

To provide a coverage at which $S_{n-1} = S_n$ it is necessary to reconfigure the network by increasing the radii of coverage and/or finding a location of sensors at which the value of the coverage area S_n is reached. In [10, 18] the approach of finding the position of the additional sensor while ensuring maximization of coverage taking into account the level of intersection of coverage areas is presented. The described approach can be used to provide $S_{n-1} = S_n$.

In the case when we managed to achieve $S_{n-1} = S_n$, remove the next sensor from the network, the coverage area will be equal to S_{n-2} , and find the network configuration for which $S_{n-2} = S_n$.

Denote by k the number of sensors for which it is not possible to reach the coverage level S_n , ie

$$S_k < S_n.$$

We introduce the concept of the limit and margin of safety of the sensor network as follows:

The limit of survivability of the sensor network is the percentage of sensors in the absence of which the required value of coverage is not achieved. This value can be calculated as follows:

$$MSSN = 1 - \frac{k}{n} * 100\%,$$

where k is the number of sensors for which it is not possible to reach the coverage level S_n , n is the total number of sensors.

The survivability margin of the sensor network is the distance (a measure of distance) between the total survivability of the network and the survivability limit that can be calculated by the formula:

$$SMSN = 100\% - MSSN.$$

Let's consider the case of data transmission in the network. We use the method proposed in [5] to estimate the limit and margin of safety of the sensor network. To do this, form a matrix of data transitions between sensors H , the elements of which will be equal to:

$$h_{i,j} = \begin{cases} 1, & \text{the connection between the sensors is present} \\ 0, & \text{there is no connection between the sensors} \end{cases} ,$$

where $i = \overline{1, n}, j = \overline{1, n}$, n is the number of sensors in the network.

Using the Dijkstra or Lee algorithm [6, 16, 17], the matrix of connections H and the matrix of importance of elements K , we find the shortest path between sensors s_1 and s_n . Denote the resulting path as s^* . Using the vector s^* we form a vector of transitions between sensors h^* .

We choose the first element of the vector $h^* h_1^*$, which is also a definite element h_{ij}^* of the matrix H , and replace its value with zero, i.e. $h_{ij}^* = 0$. After replacement, find the path between sensors s_1 and s_n . If the path cannot be found, the selected connection between the sensors is vulnerable. Next, return the element h_{ij}^* to the previous value and select the next element of the vector h^* . The final assessment of the vulnerability of the network to loss of connections will be calculated as:

$$c_h = 1 - \frac{c_0}{m_t},$$

where c_0 is the number of negative results of finding the path, m_t is the number of elements in the vector h^* .

To estimate the survivability of the network relative to the loss of sensors, take the second element of the vector s^* and replace with zero the elements of the corresponding row and column of the matrix of connections H . After replacement, find the path from s_1 to s_n . If it is impossible to find the path, the selected sensor is vulnerable. In the next step we return the previous values to the elements of the matrix H , select the next element of the vector s^* and repeat the procedure. The final estimate of the survivability of the network in case of loss of sensors is calculated by the formula:

$$c_s = 1 - \frac{s_0}{m_s - 2},$$

where s_0 is the number of negative results of finding the path, m_s is the number of elements in the vector s^* .

In this formula, the denominator is defined as $m_s - 2$ because the initial and final sensors are not taken into account.

The final estimate is calculated as the probability of simultaneous occurrence of two independent events [5] by the formula:

$$c = c_s c_h.$$

Let's consider the case of different bandwidth sensors and communication channels. Let's look at the differences in the throughput of the sensors and the channels of the connection. To assess the survivability of the cell, use the sensors in either the channels or the link, to ensure the greatest throughput of the building, two vectors P_{max}^s and P_{max}^h are formed. The elements of these vectors will be sensors and links between sensors, which will ensure the maximum flow of data in the measure. For the formation of these vectors, it is possible to use the Ford-Fulkerson algorithm [16] and the significant value of the maximum flow as P_{max} . The molding algorithm for assessing the survivability of the treadmill in the fall of the elements, which ensures the maximum flow

is similar to the one described above, but it is less important to ensure the maximum flow in the treadmill. Obviously, the result of the robotic algorithm will be two vectors: h_{weak}^{max} is the vector of different sensors and c_{weak}^{max} is the vector of different sounds. Residual scores can be calculated by looking at

$$c_p^h = 1 - \frac{c_0^h}{m_p^h},$$

$$c_p^s = 1 - \frac{c_0^s}{m_p^s - 2},$$

where c_0^h is the number of vulnerable connections, m_p^h is the number of elements that provide maximum flux (connection vector P_{max}^h), c_0^s is the number of vulnerable sensors, m_p^s is the number of elements in the vector of P_{max}^s , providing maximum data flow in the network.

Taking into account the estimates of network survivability in case of loss of elements that ensure the maximum flow of data in the network, the final estimate can be modified as follows

$$c = c_s c_h c_p^h c_p^s.$$

Since the value of $c = 1$ means the survivability of the network, the limit and margin of safety of the sensor network will be the values:

$$MSSN_e = \left(1 - \frac{1}{c}\right) * 100\%,$$

$$SMSN_e = 100\% - MSSN_e.$$

In general, the algorithm to ensure the survivability of the coverage area consists of three steps:

1. Checking the level of coverage of the territory and ensuring maximization of coverage in the case of:

$$\frac{T}{S_n} < 1,$$

where T is the area of the territory, S_n is the coverage area of the network consisting of n sensors.

Coverage maximization is achieved by connecting additional sensors to the network [9, 25];

2. Identification of vulnerable sensors using the proposed approach;
3. Ensuring the viability of the network through redundancy. By redundancy we mean additional sensors and communication channels between sensors for the required defined survival limit \overline{M} :

$$MSSN_e \geq \overline{M}.$$

The third step is an iterative procedure to ensure survivability for each vulnerable sensor, followed by verification of overall survivability. Schematically described above algorithm can be represented as follows (Fig. 2):

Obviously, the consequence of the surplus is an increase in resource and energy consumption. To ensure energy efficiency, we will have duplicate sensors

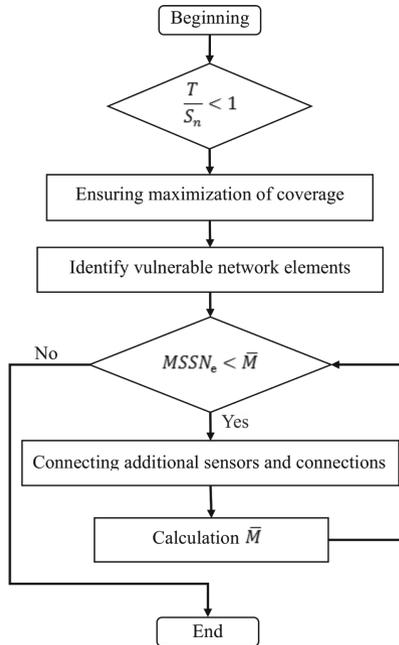


Fig. 2. Algorithm for ensuring the survivability of the sensor network

with the minimum allowable coverage radius and communication only with the main sensor. The backup sensor with a certain frequency t_i polls the main sensor and turns on only when it will not receive a response from the main sensor during the time $t_{silence}$.

3.4 Computer Simulation, Results and Discussion

Let's consider a network consisting of 6 sensors with the following parameters (Table 2):

Table 2. Characteristics of sensors

Position	Coverage radius, m
(35;40)	10
(17;30)	8
(20;45)	10
(50;55)	13
(15;15)	10
(35;15)	17

Schematically, this network can be represented as follows (Fig. 3):

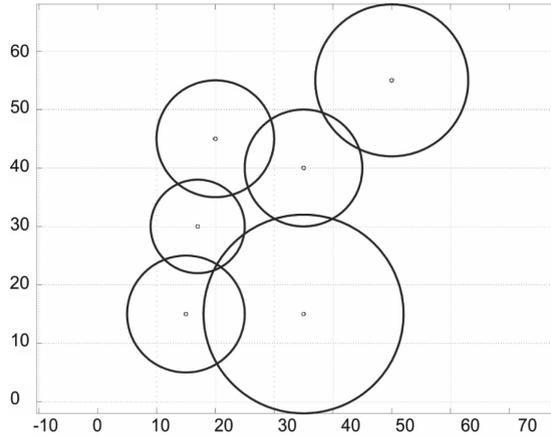


Fig. 3. Schematic representation of the incoming network

Using the approaches proposed above, we will determine the limit, the margin of safety of the network and find vulnerable sensors. The following results were obtained: $SMSN_e = 66\%$,

$$SMSN_e = 34\%$$

vulnerable sensors: (35; 40) and (35; 15).

By adding additional sensors, the resulting network will look like Fig. 4. A schematic representation of the network using minimization of energy consumption is presented in Fig. 5. Additional sensors are shown by a dashed line.

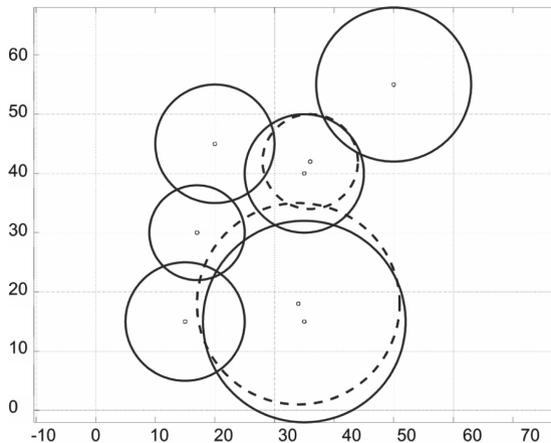


Fig. 4. The result of ensuring the survivability of the network without optimizing energy consumption

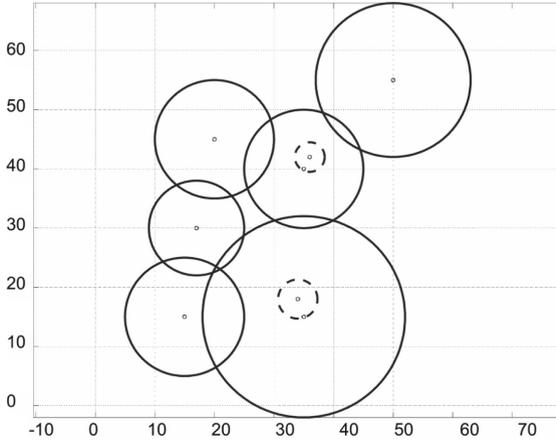


Fig. 5. The result of ensuring the survivability of the network with optimization of energy consumption

Let us calculate the energy consumption in mAh and apply the approach proposed in [15]. The values of power consumption, limit and margin of survival, as well as the number of life cycles of the network relative to the initial parameters are shown in Fig. 6. The power consumption of the input network is 4624 mAh, the resulting network without power consumption optimization is 8649 mAh, the resulting network with power consumption optimization is 5446.4 mAh. When optimizing energy consumption, the resulting network consumes 1.58 times less energy.

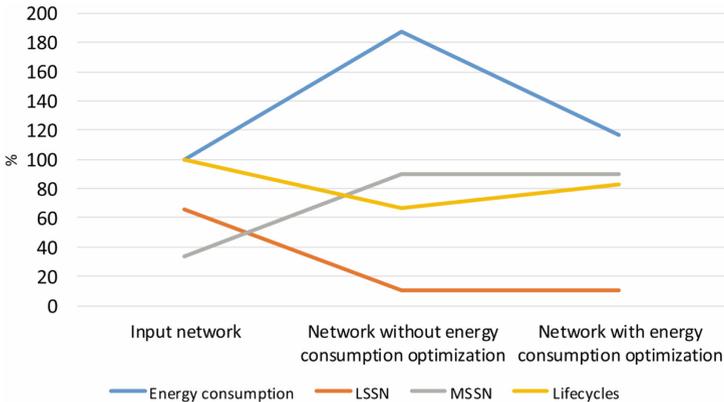


Fig. 6. The value of the resulting and initial characteristics of the network

4 Conclusions

The authors propose signs and indicators of functional stability of the data exchange system. The concepts of boundary and margin of functional stability are introduced. Quantitative methods for assessing functional stability based on the above indicators are proposed.

The paper considers the survivability of sensor networks. Survivability is considered as a component of functional stability of networks. The description of development of the basic concepts and applied approaches of functional stability on examples of maintenance of functional stability of concrete kinds of technical systems is resulted. The survivability of the sensor network is presented as the property of the network to provide coverage and data transmission in case of loss or failure of network components such as sensors and connections between sensors. The formation of the stock and the limits of survivability of sensor networks are described. These values are percentages and are formed in cases of maximum coverage and in cases of data transmission in the network. An algorithm for ensuring the survivability of sensor networks is presented. Ensuring survivability is achieved by adding additional sensors to the network and connections between sensors. The approach of optimization of energy consumption of the network with additional sensors by adjusting the radii of coverage of additional sensors depending on the operating condition of the main sensors is presented. The results of computational algorithms confirm the effectiveness of the proposed approaches. During computational experiments, the survivability limit of the input sensor network was increased by 34%. The proposed method of optimizing the energy consumption of additional sensors has reduced the energy consumption of the resulting network by 1.58 times.

The creation of functionally stable information and telecommunication systems will increase the efficiency of operation in the event of emergencies in the event of technical violations, failures and the impact of external factors.

A promising area of further research is the improvement of algorithmic and software information and telecommunications systems for testing actions in the context of outrageous factors. The issue of parrying abnormal, emergency situations in information and telecommunication systems in real time has become of paramount importance and needs scientific and practical justification.

References

1. Albin, L.C.P., Duarte, E.P., Ziwich, R.P.: A generalized model for distributed comparison-based system-level diagnosis. *J. Braz. Comput. Soc.* **10**(3), 42–54 (2005). <https://doi.org/10.1007/BF03192365>
2. Babichev, S., Taif, M.A., Lytvynenko, V.: Inductive model of data clustering based on the agglomerative hierarchical algorithm. In: Proceedings of the 2016 IEEE 1st International Conference on Data Stream Mining and Processing, DSMP 2016, pp. 19–22 (2016). <https://doi.org/10.1109/DSMP.2016.7583499>
3. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopolymers Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>

4. Barborak, M., Malek, M., Dahbura, A.: The consensus problem in fault-tolerant computing. *ACM Comput. Surv.* **25**(2), 171–220 (1993). <https://doi.org/10.1145/152610.152612>
5. Bianchini, R., Buskens, R.: An adaptive distributed system-level diagnosis algorithm and its implementation. In: 21st International IEEE Symposium on Fault-tolerant Computing, pp. 222–229 (1997). <https://doi.org/10.1109/FTCS.1991.146665>
6. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Dijkstra’s algorithm. MIT Press and McGraw-Hill, *Introduction to Algorithms* (2001)
7. Fujiwara, H., Kinoshita, K.: Some existence theorems for probabilistically diagnosable systems. *IEEE Trans. Comp.* **C-27**(4), 297–303 (1981). <https://doi.org/10.1109/TC.1978.1675111>
8. Jarrah, H., Sarkar, N., Gutierrez, J.: Comparison-based system-level fault diagnosis protocols for obile ad-hoc networks. *J. Netw. Comput. Appl.* **60**, 68–81 (2016). <https://doi.org/10.1016/j.jnca.2015.12.002>
9. Javaid, M., Haleem, A., Rab, S., Singh, R.P., Suman, R.: Sensors for daily life: a review. *Sensors Int.* (2) (2021). <https://doi.org/10.1016/j.sintl.2021.100121>
10. Klenke, A.: *Probability Theory: A Comprehensive Course*. Springer Science & Business Media (2013)
11. Mahapatro, A., Khilar, P.: Fault diagnosis in wireless sensor networks. *IEEE Commun. Surv. Tutorials* **15**, 2000–2026 (2013). <https://doi.org/10.1109/SURV.2013.030713.00062>
12. Mashkov, V.: New approach to system level self-diagnosis. In: 11th IEEE International Conference on Computer and Information Technology, CIT 2011, CIT 2011, pp. 579–584 (2011). <https://doi.org/10.1109/CIT.2011.12>
13. Mashkov, V., Mashkov, O.: Interpretation of diagnosis problem of system level self-diagnosis. *Mathematical Modeling and Computing* **2**(1), 71–76 (2015)
14. Mashkov, V.A., Mashkov, O.A.: Interpretation of diagnosis problem of system level self-diagnosis. *Math. Modeli. Comput.* **2**(1), 71–76 (2015)
15. Mykus, S.A., Bondarenko, V.E., Rudenko, N.V.: Models of synthesis of functionally stable computer network. *Sci. Pract. J. Commun.* (6), 29–32 (2014)
16. Neto, E., Callou, G.: An approach based on ford-fulkerson algorithm to optimize network bandwidth usage. In: Brazilian Symposium on Computing Systems Engineering (SBESC), pp. 76–79 (2015). <https://doi.org/10.1109/SBESC.2015.21>
17. Petrivskiy, V., Dimitrov, G., Shevchenko, V., Bychkov, O., Garvanova, M., Panayotova, G., Petrov, P.: Information technology for big data sensor networks stability estimation. *Inf. Secur.* **47**(1), 141–154 (2020). <https://doi.org/10.11610/isij.4710>
18. Polanczyk, M., Strzelecki, M., Slot, K.: Lee-algorithm based path replanner for dynamic environments. *Signals Electron. Syst. (ICSES)* (2012). <https://doi.org/10.1109/ICSES.2012.6382245>
19. Qin, L., He, X., Zhou, D.: A survey of fault diagnosis for swarm systems. *Syst. Sci. Control Eng.* **2**, 13–23 (2014). <https://doi.org/10.1080/21642583.2013.873745>
20. Quaid, E.: *Analysis of complex systems*. M.: Sov. radio (1969)
21. Reliability of technology: Terms and definitions: DSTU 2860–94 (1994). <https://dnaop.com/html/2273.3.html>
22. Shevchenko, A.V., Bychkov, A.S., Dimitrov, G.P., Shevchenko, V.L.: Perfection of computer epidemic model by estimation of functional stability of the information system. *J. Autom. Inf. Sci.* **52**(1), 29–40 (2020). <https://doi.org/10.1615/JAutomatInfScien.v52.i1.40>

23. Shevchenko, A.V., Bychkov, A.S., Novotna, V., Shevchenko, V.L.: Improvement of the model of computer epidemics based on expanding the set of possible states of the information systems objects. *J. Autom. Inf. Sci.* **15**(11), 34–39 (2019). <https://doi.org/10.1615/JAutomatInfScien.v51.i11.40>
24. Somani, A.: System Level Diagnosis: A Review (1997). <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.9488>
25. Theo, S.: The Latest In Sensors and Applications (2020). <https://www.electronicsforu.com/technology-trends/tech-focus/latest-sensors-applications>
26. Weber, A., Kutzke, A.R., Chessa, S.: Energy-aware test connection assignment for the self-diagnosis of a wireless sensor network. *J. Braz. Comput. Soc.* **18**(1), 19–27 (2012). <https://doi.org/10.1007/s13173-012-0057-7>



Adaptive Decision-Making Strategies in the Game with Environment

Petro Kravets¹, Victoria Vysotska^{1,2}, Vasyl Lytvyn¹,
and Lyubomyr Chyrun³

¹ Lviv Polytechnic National University, Lviv, Ukraine

{Petro.O.Kravets,Victoria.A.Vysotska,Vasyl.V.Lytvyn}@lpnu.ua

² Osnabrück University, Osnabrück, Germany

Victoria.Vysotska@uni-osnabrueck.de

³ Ivan Franko National University of Lviv, Lviv, Ukraine

Lyubomyr.Chyrun@lnu.edu.ua

Abstract. The problems that occurred during decision-making and supported by the stochastic game model with the environment are investigated and analysed. A method of adaptive choice of solutions based on stimulatory learning and application of Boltzmann distribution is developed. The algorithm and software of the decision-making system in the games with the environment are developed. The computer simulation results of stochastic choice of solutions are analysed and described. The developed decision-making method provides stochastic minimisation of the agent's mean loss function and is based on adaptive parametric identification of the environment using Q-learning and defining a mixed strategy on the Boltzmann distribution. With proper parameter selection, the method provides close to 1 order of asymptotic convergence, the limit value for a given class of recurrent methods.

Keywords: Decision making · Stochastic nature play · Adaptive strategies · Boltzmann distribution · Q-learning · Cybernetics development

1 Introduction

There is a rapid development of computer and information systems in all human activity spheres in the modern world. The task is to help the persons as much as possible or replace them when performing routine or potentially dangerous operations. Special software agents for information systems and robots for physical systems are developing. They can perceive and process information images of the environment and make decisions independently. The robotic systems process information more accurately and reliably by computer implementation, without fatigue and unnecessary emotions. Computer information technology dynamics have led to a new wave formation of scientists and engineers' interest in developing and implementing cybernetic systems for various purposes, such as commercial, security, linguistical, medical, intelligence, social, industrial, household etc.

The natural environment is the inspiration for cybernetics development as a general laws science for information management, collection, storage, processing and transmission in intelligent systems. Thus, the American mathematician Norbert Wiener pointed to the similarity of control processes in living organisms and complex technical systems in his work "Cybernetics: or, Control and Communication in the Animal and the Machine" (1948). The borrowing of natural environment methods to develop engineering devices and techniques led to new science such as bionics (1960) at the intersection of cybernetics, biophysics, biochemistry, engineering psychology, and other sciences. The task of bionics is related to the principles and patterns study of functioning living organisms for use in artificially organised systems. Their need is due to the extreme efficiency and perfection of natural forms polished during evolution. It is believed that the living organism's development is guided by an adaptive method of trial and error with the best selection and more resistance to survival forms [5]. Most of the practical applications for bionics are related to the physicochemical properties of individual living organisms. Equally important is studying the behavioural model of living beings' self-organisation in a changing existence environment. These models transfer into technical and information systems for data management or decision-making. Self-organisation is a purposeful spatial-temporal process for creating, organising or improving the structure and functions of a complex dynamic system due to its internal factors without external influence. The constructive mechanism for complex systems self-organisation is multivariate, alternative, randomness and selectivity of possible actions of active elements.

External manifestations of self-organisation are the ordered structure formation in the patterns formed at the macro-level and result from local interactions of the system's constituent elements at the micro-level. In addition to the orderly distribution of structural elements, there are often patterns for individual living organisms and their populations' behaviour. Behavioural patterns determine the form of ordered actions for system elements, which arise under certain conditions from chaos. Genetic, immune, flock, swarm, and ant algorithms are used to construct behavioural patterns of self-organisation. In addition, the study of behavioural patterns also uses methods of enhanced Q-learning, networks of automata, artificial neural networks, hidden Markov networks, and deterministic or stochastic games. Computer simulations for a living population's behaviour pattern are performed using artificial agent systems. An agent is an autonomous information software that processes signals from the environment, other agents, and a human for possible action selection or making decisions to achieve goals.

The agent behaviour pattern is a reproducible response to environmental stimuli. It is sometimes up to the action's isomorphism. It is aimed at achieving a specific goal in multi-step adaptive learning. The question arises about how complex the agent's organisation must be to reproduce a living organism's behaviour in the environment's uncertainty. Alternatively, the complex system for processing the environment's reactions must learn to exist effectively and survive in a changing world.

The adaptive methods are used for making decisions in uncertain conditions [5, 11]. It allows optimising the options choice for action in the following time moments based on the proper processing of current information about the system. Decision-making in situational uncertainty conditions is carried out by repeated operations from a finite set of possible options. The target solution search cannot be performed by a single-stage optimisation method or a sequential search method of all possible solutions and system states because the system may give a different signal response for the same option at other times [4, 13]. In addition, repeating the steps makes it possible to gather the necessary information about the system and compensate for its uncertainty.

In uncertain conditions, the individual decision-making process is considered a stochastic play model with nature [16]. An intellectual agent and the natural environment take part in the game. An agent is a model of a decision-maker. The natural environment models the problem area of decision-making. Both players are included in the feedback loop. The agent selects and implements one of the action options with the appropriate environment response at successive time moments. The agent performs an adaptive adjustment of the strategy for choosing action options when analysing the signal from the environment [3, 10]. Adjustments are made to ensure the agent's maximum gain or minimum loss throughout the decision-making process [6, 14]. Constructing patterns for game behaviour of agents in uncertain conditions is not comprehensively covered in the related professional and scientific works. Therefore, the game model research of the adaptive strategies formation for agent's behaviour is an urgent scientific and practical problem [4, 11].

The main contributions of this paper are summarized as follows:

- The paper investigates and analyses the problems that occurred during decision-making and is supported by the stochastic game model with the environment.
- A method of adaptive choice of solutions based on stimulatory learning and application of Boltzmann distribution is developed and described.
- The algorithm and software of the decision-making system in the games with the environment are developed and described. The developed decision-making method provides stochastic minimisation of the agent's mean loss function. This method is based on adaptive parametric environment identification using Q -learning and defining a mixed strategy on the Boltzmann distribution.
- The paper analyses and describes the experiment results of stochastic choice of solutions. The method convergence is ensured by observing the primary stochastic approximation conditions. The convergence parameters are theoretically determined and refined experimentally. The technique provides close to 1 order of asymptotic convergence with proper parameter selection. It is the limit value for a given class of recurrent methods.
- The developed decision-making method provides stochastic minimisation of the agent's mean loss function and is based on adaptive parametric identification of the environment using Q -learning and defining a mixed strategy on the Boltzmann distribution. With proper parameter selection, the method

provides close to 1 order of asymptotic convergence, the limit value for a given class of recurrent methods.

The adaptive method for forming decision-making agent behaviour strategies in the game with the environment in this work is investigated. The behavioural patterns reproduction based on the stochastic game model reveals the causes and mechanisms for their formation in nature, better help to understand the natural evolution laws and helps to apply game algorithms in practice for various decision support systems construction in bioinformatics. The proposed method development is possible by considering the state dynamics of the decision-making environment and the transition from a reactive to a cognitive model of the agent.

2 Related Works

Adaptive methods [4,5,11] are used to make decisions under uncertainty. They allow us to optimise the chosen options at the following moments based on the proper processing of current system data, for example, for prognosis or diagnosis by symptomatology. The construction of adaptive decision-making methods is performed based on random processes. The decision-making process is considered a stochastic game model with a natural environment. An intelligent agent and the natural environment play the game. The agent is the decision-maker model. The natural environment models the problematic area of decision-making. Players are included in the feedback loop. The agent selects and implements one of the appropriate responses to the environment at successive times. The agent performs an adaptive adjustment of choosing options for the following time moment, analysing the signal from the environment. Adjustments ensure the agent's maximum gain or minimum loss throughout the decision-making process. The first studies of stochastic games related to analysing automaton games with the environment [3,6,10,13,14,16]. Different tasks are made depending on:

- Automatic machine type (deterministic or random; with a constant or variable structure; with a limited or unlimited number of states; with discrete or continuous system states);
- Random environment type (stationary, non-stationary, with transition states; finite or boundless Markov chain);
- Current winnings or losers' values: discrete or continuous; limited or unlimited; scalar or vector.

These problems investigate the self-learning automatic device's behaviour rationality in random environments. It mostly boils down to asymptotic minimisation of the average penalty for incorrectly chosen actions.

We use the following method to study the environment's behaviour games:

- The chains Markov theory;
- Diffusion;
- Approximation;

- Martingales;
- Random search;
- Potentials;
- Factorisation;
- Laplace transform;
- Nonlinear programming;
- Numerical methods.

Automatons' group games research in random environments after obtaining the first results on individual machine dynamics is launched as a meaningful model of collective behaviour [6,16]. M.L. Tsetlin supported work in this direction. He believed that complex behaviours of objects are implemented by finite-state machine sets that behave rationally in random environments.

Multiple finite-state machine games are primarily investigated by computer simulation. The general case of the N individuals' game was considered in [18]. These game methods were obtained using stochastic approximation, gradient projection and regularisation methods.

Stochastic game models are mainly used to solve decision-making under uncertainty in biology, psychology, sociology, political science, military affairs, economics, ecology, technical systems, etc. [13,16]. Game theory has some unresolved problems despite a considerable period of development. The current research relevance in game theory is confirmed, for example, by the extensive list of bibliography Internet sources.

The International Society for Dynamic Games coordinates research in the game theory field (<http://www.hut.fi/HUT/Systems.Analysis/isdg/>).

The main problems of game theory are outlined at the Nobel Symposium on Game Theory (June 18–20, 1993, Björkborn, Sweden) and some International Symposiums on Dynamic Games and Applications. Identified promising areas of research for games include:

- Repetitive games;
- Network games in telecommunications and transportation;
- Evolutionary games;
- Pursuit and escape games;
- Dynamic games in economics and management;
- Games training and adaptation;
- Stochastic games;
- Games in finance and marketing;
- Managing the environment, energy and resources;
- Trading games;
- Cooperative equilibrium games;
- Numerical methods and computer implementations of game models.

Many of the leading scholars in game theory focus on studying stochastic evolutionary and dynamic repetitive games. The most critical issue is the self-learning game methods development and research. There is a shift from behavioural

search to cognitive concepts of developing self-learning game methods. Mathematical methods and simulation tools for stochastic games are fundamental techniques for studying multi-agent intelligent systems [15, 17]. The optimality criteria research for the collective coexistence of active elements (agents) of homogeneous and heterogeneous distributed management and decision-making systems continues.

Applied game theory has shifted its initial cognitive interests in biology to pragmatic economics, management, and marketing challenges [9, 12, 15, 17]. However, the powerful mathematical game theory methods have not yet found the proper place when solving technical problems [1, 2, 7, 8].

Known game methods of solution choices are mainly based on hybrid strategies to optimise average winnings' functions. Environment information is available to players in the form of current winnings or losers [7]. This paper proposes using existing software to adapt decision-making parameters identification and definition of mixed strategies based on the Boltzmann distribution. The paper aims to develop an effective decision-making method in uncertainty based on parametric identification in the adaptive game of the agent with the environment.

3 Materials and Methods

Consider the model of the decision-making system (S, A) . It consists of the feedback loop included in the environment S and agent A [17]. An agent is an active intellectual system of decision-making and implementation. The agent's interaction with the environment is described in the game with the natural environment. The natural environment $S = (U, \xi, F)$ is given by the input vector

$$U = (u(1), u(2), \dots, u(N)),$$

where scalar output $\xi \in \mathbb{R}^1$ and transfer function

$$F : u \rightarrow \xi.$$

Let the process $F = F(v(u), d(u))$ is generated a random variable generator ξ with mathematical expectation $v(u)$ and variance $d(u)$ at $\forall u \in U$. The environment outputs are the agent's choices estimates.

The agent $A = (\xi, U, \Pi)$ is given a scalar input ξ (corresponding to the natural environment output), a vector of pure strategies U (matching the environment inputs), and a rule of pure strategies choosing

$$\Pi : \xi \rightarrow u.$$

Pure strategies determine discrete options. Environment settings:

$$v = (v_1, v_2, \dots, v_N),$$

$$d = (d_1, d_2, \dots, d_N)$$

and the distribution functions F of a priori unknown agent appearance. The choices are made at discrete times $n = 1, 2, \dots$. The agent option $u_n = u \in U$, after selecting, observes a random implementation $\xi_n(u_n) \sim F(v(u_n), d(u_n))$ of the parameter $v(u_n)$. The value $\xi_n(u_n)$ is interpreted as the current loser of the chosen agent u_n . It is considered that accidental losers $\{\xi_n\}$ are independent $\forall u_n \in U$ at $n = 1, 2, \dots$ and have constant expectations

$$M\{\xi_n(u)\} = v(u) = \text{const}$$

and limited second moment

$$\sup_n M\{[\xi_n(u)]^2\} = \sigma^2(u) < \infty.$$

The average loss of the agent at a time n is defined as:

$$\Xi_n(\{u_n\}) = \frac{1}{n} \sum_{t=1}^n \xi_t. \tag{1}$$

The purpose of the agent is to minimise the averaged loser’s function:

$$\overline{\lim}_{n \rightarrow \infty} \Xi_n \rightarrow \min. \tag{2}$$

To solve problem (2), we must define a rule Π sequence formation $\{u_n\}$ variants of decisions in time. We will execute forming a sequence of solutions $\{u_n\}$ with suitable properties based on a dynamic mixed strategy

$$p_n = (p_n(1), p_n(2), \dots, p_n(N)).$$

Elements $p_n(j)$, $j = 1..N$ of mixed strategy vectors are conditional probabilities of choosing pure strategies depending on the current solution and the obtained result. The hybrid strategy takes on a unit simplex value [18]:

$$S^N = \{p | \sum_{j=1}^N p(j) = 1; p(j) \geq 0(j = 1..N)\}.$$

Decision-making rule: $\Pi : p_n \xrightarrow{\xi_n} p_{n+1} \xrightarrow{\omega} u_{n+1}$ should ensure the mixed strategy point movement p_n on a unit simplex in the intermediate direction for optimal resolution under the action of current losers ξ_n and selecting an option u_{n+1} based on a hybrid strategy p_{n+1} realisation of a chance ω . A recurrent transformation series of the mixed-strategy vector for solving the problem (2) is given [18].

This article proposes a new method of recurrent choice for decision-making based on modified Markov Q -learning [15]:

$$Q_{n+1}(u_n) = Q_n(u_n) - \gamma_n[\xi_n(u_n) + Q_n(u_n)], \tag{3}$$

where $Q_n(u_n) \in \mathbb{R}^1$ is the environment S estimation function parameter $v(u_n)$ for the selected at a time n option $u_n = u \in U$; $\gamma_n > 0$ is the parameter that

regulates the methods' step size; $\xi_n(u_n) \in \mathbb{R}^1$ is the current loser of the chosen agent u_n .

The parameter γ_n is a monotonically decreasing value of actual type and is calculated as:

$$\gamma_n = \gamma n^{-\alpha}, \tag{4}$$

where $\gamma > 0$ and $\alpha > 0$.

Method (3) performs adaptive parametric identification of the decision environment depending on the selected variant u_n and the size of the current loser ξ_n .

The Boltzmann distribution determines the elements of a mixed strategy:

$$p(u) = \frac{e^{\frac{Q(u)}{T}}}{\sum_{\alpha \in U} e^{\frac{Q(\alpha)}{T}}}, \forall u \in U. \tag{5}$$

where T is the temperature parameter of the system.

Conversion (5) ensures that a mixed strategy is appropriate p for a unit simplex S^N . Condition (5) determines the corresponding value of the pure strategies:

$$u_n = \{u(k) | k = \arg \min_k \sum_{i=1}^k p_n(i) > \omega(k = 1..N)\}, \tag{6}$$

where $\omega \in [0, 1]$ is a random variable with uniform distribution.

Convergence $\lim_{n \rightarrow \infty} \| p_n - p^* \| \rightarrow 0$ method (3)–(6) to the optimal strategy, which minimises the mean loss (1), is provided by the general stochastic approximation conditions [9]:

$$\sum_{n=0}^{\infty} \gamma_n = \infty,$$

$$\sum_{n=0}^{\infty} \gamma_n^2 < \infty.$$

The quality of decision-making is judged by the deviation $\delta_n = \| p_n - p^* \|^2$ of the current mixed strategy p_n from the optimal strategy p^* and the time-averaged deviation:

$$\Delta_n = \frac{1}{n} \sum_{t=1}^n \delta_t. \tag{7}$$

Algorithm 1: The Decision Options Choice

Initialization:

Step 1. Set the initial parameter values (Fig. 1):
 N – is the number of pure strategies;
 $v = (v_1, \dots, v_N)$ – is the mathematical expectations vector of the losing agent;
 $d = (d_1, \dots, d_N)$ – is the dispersion vector of the losing agent;
 $U = (u_1, \dots, u_N)$ – is the pure strategies’ values vector;
 $p = (\frac{1}{N}, \dots, \frac{1}{N})$ – is the hybrid strategy (probabilities vector for choosing pure strategies);
 $T > 0$ – is the temperature parameter of Boltzmann distribution;
 $Q_0(u) = 0, \forall u \in U$ – is environment identification parameters;
 $\gamma > 0$ – is the learning step;
 $\alpha \in (0, 1]$ – is the learning steps order;
 ε – is average losses calculation accuracy;
 $\xi \sim Z(v, d)$ – is the current winnings distribution law;
 n_{max} – is the maximum number for method steps;
 $n = 0$ – is the starting time moment.

while $n < n_{max}$ *or* $\Delta_n \geq \varepsilon$ **do**

Step 2. Choose the solution $u_n \in U$ based on (6).
 Step 3. Get the current loser value ξ_n .
 Step 4. Calculate parameter value γ_n (4).
 Step 5. Calculate parameter value $Q_n(u_n)$ by to (3).
 Step 6. Compute vector elements of mixed strategy p_n based on (5).
 Step 7. Calculate the characteristics of the decision-making quality δ_n and Δ_n by to (7).
 Step 8. Set the next time moment $n := n + 1$.

end

4 Experiments and Results

Verification of the efficiency (ability to minimise average loss) of the proposed method is performed by simulation software modelling. The environment is entrances set $N = 4$ and one exit ξ . Stochastic conversion of inputs to the output signal is carried out by normal distribution $\xi \sim Normal(v_i, d_i), i = 1..N$ with vectors of a priori unknown parameters $v = (0.5, 0.9, 0.1, 0.7)$ and dispersions $d = (d_1, \dots, d_N)$. The lowest average losing agent for a given v optimal strategy $p^* = (0, 0, 1, 0)$ is determined.

Normally distributed random variables (Gaussian law) are calculated using the sum of twelve evenly distributed segments $[0, 1]$:

$$\xi_n(u, \omega) = v(u) + \sqrt{d(u)} \left(\sum_{j=1}^{12} \omega_j - 6 \right),$$

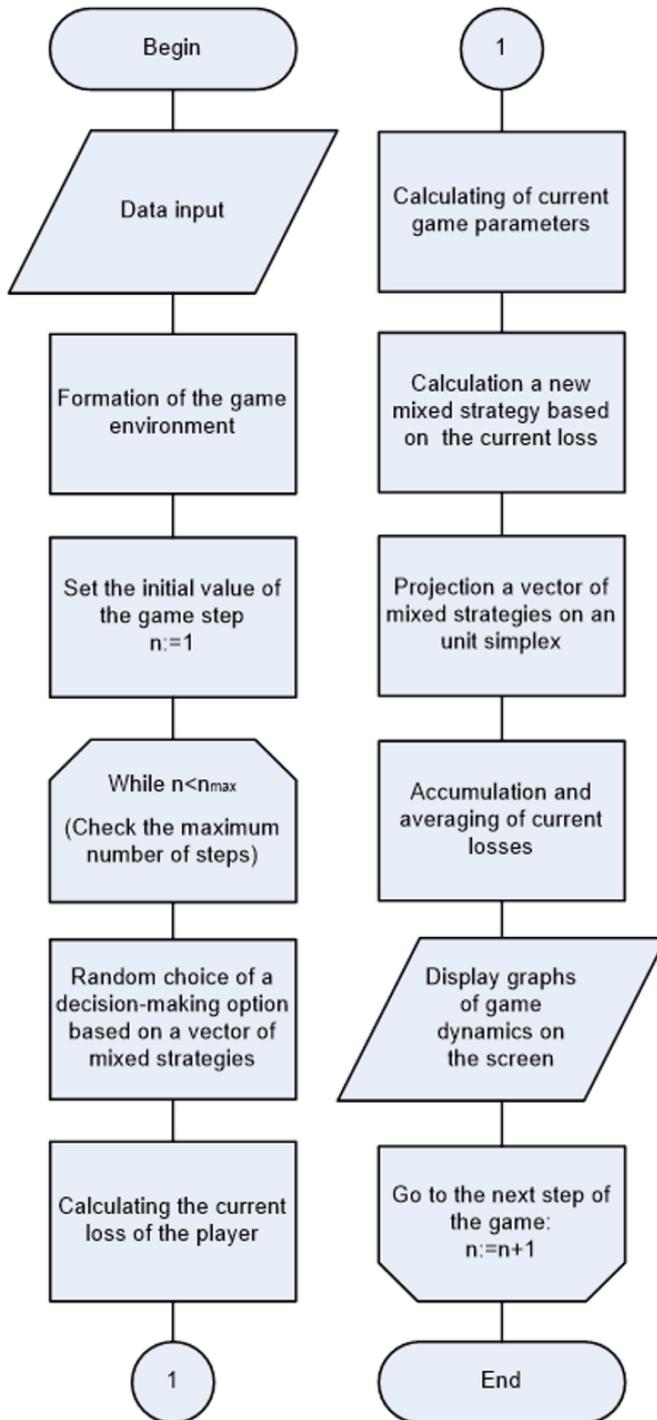


Fig. 1. Flow chart of the stochastic game algorithm

where $u \in U$; $\omega \in [0, 1]$ - is a valid random number with a uniform distribution law.

Binary losers $\xi_n(u, \omega) \in \{0, 1\}$ are calculated with probabilities $v(u) \in [0, 1]$, $\forall u \in U$:

$$\xi_n(u, \omega) = \begin{cases} 0, & \omega > v(u) \\ 1, & \omega \leq v(u) \end{cases}$$

At the moment, n the agent receives the value of the loser ξ_n . According to (3) - (6), it implements one of N the pure strategies in method (3), the parameter

$$\gamma_n = \gamma n^{-\alpha},$$

where $\gamma = 1$, and the initial value $Q_0(u) = 0, \forall u \in U$.

The estimation of the asymptotic order of convergence velocity is performed by the Zhong method [9]:

$$\overline{\lim}_{n \rightarrow \infty} n^\theta M\{\Delta_n\} \leq \vartheta, \tag{8}$$

where θ is parameter order; ϑ is the convergence rate magnitude; Δ_n is the time-averaged Euclidean rejection rate of the current mixed strategy p_n from the optimal value p^* .

More θ and less ϑ corresponds to a more incredible speed of convergence of the game method. The length of the test sample is 10 thousand steps. The process behaviour Δ_n in time is approximated by dependency based on the convergence rate estimation (8):

$$\Delta_n = \frac{\vartheta}{n^\theta},$$

where $\vartheta > 0, \theta \in (0, 1], n = 1, 2, \dots$

We obtain a linear relation after the logarithm:

$$\lg \Delta_n = \lg \vartheta - \theta \lg n. \tag{9}$$

In addition, we get the dependency:

$$\lg \Delta_n = f(\lg n).$$

The parameter

$$\theta = \frac{\lg \Delta_n}{\lg n}$$

indicates the test method convergence order.

A random process approximation is performed $\lg \Delta_n$ with linear dependence (9) on the segment $\lg n \in [3, 4]$ in steps of 0.1 using the least-squares method to determine the convergence speed order θ . Smoothing the random component of the convergence rate and allocating the speed order is performed by averaging over the random process realisations Δ_n . The simulation results are shown in Figs. 2, 3, 4, 5 and 6 on a logarithmic scale. Figure 2–Fig. 3 shows the graphs of the middleware function Ξ_n and average norm function Δ_n deviation of the

current mixed strategy from the optimal value for different variance values d of normal distribution of current losers. Obtained data is for the value $T = 0.01$ temperature coefficient of the Boltzmann distribution.

The reduction in time of the mean loss function to the minimum value $v_3 = 0.1$ and the decrease in the Euclidean norm of the difference between mixed and optimal strategies vectors indicate the efficiency of the developed recurrent method. The variance increases lead to a slight decrease in the magnitude of the convergence rate ϑ and have little or no effect on the order θ . It is estimated by the slope tangent of the linear approximation of the function Δ_n with the moments' axis. For a given T order of convergence speed θ approaching 1.

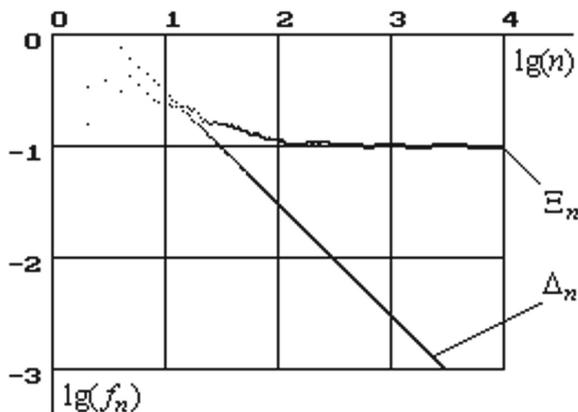


Fig. 2. Method convergence characteristics at $d = 0.01$

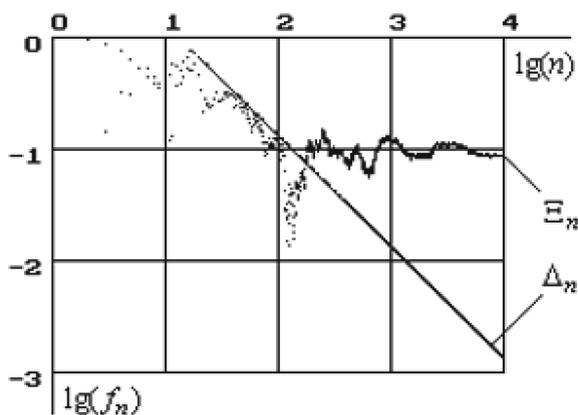


Fig. 3. Method convergence characteristics at $d = 1$

5 Discussion

Figure 4 shows the convergence characteristics of methods (3)–(6) for binary losers $T = 0.01$. It is experimentally established that changing the law of distribution of current losers (for example, binary, uniform, exponential, etc.) does not significantly affect the order of convergence speed of the developed method. Significant influence on the convergence rate of the process (3)–(6) has the Boltzmann temperature distribution coefficient value. Figure 5 shows the error function graphs Δ_n obtained for the variance value $d = 0.01$ and different temperature coefficient values T .

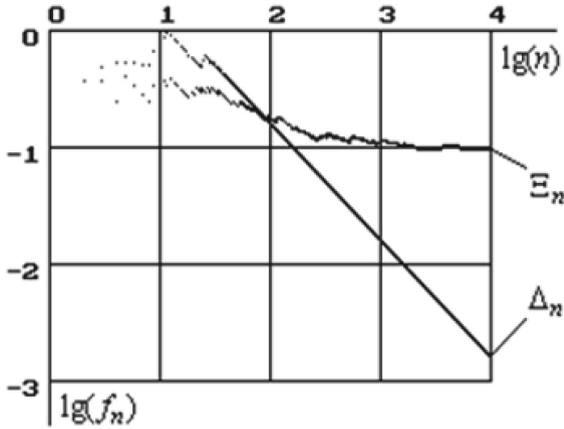


Fig. 4. Method convergence characteristics for binary losers

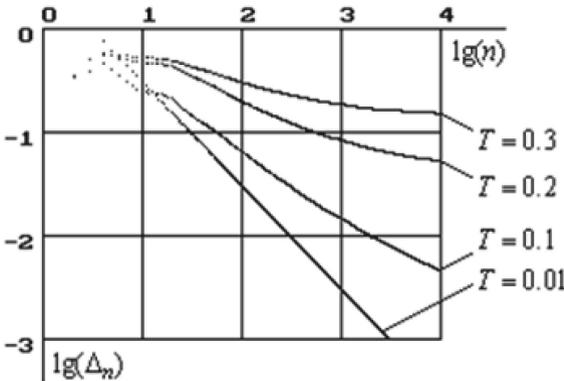


Fig. 5. Temperature coefficient effect on method convergence

When we decrease the parameter T , the convergence velocity of method (3)–(6) increases because at small T is close to the “greedy” [7, 12]. And, at large, T

is close to the uniform law of solutions choice. The developed method is resistant to the abrupt change of environmental parameters, confirmed by the ones shown in Fig. 6 results.

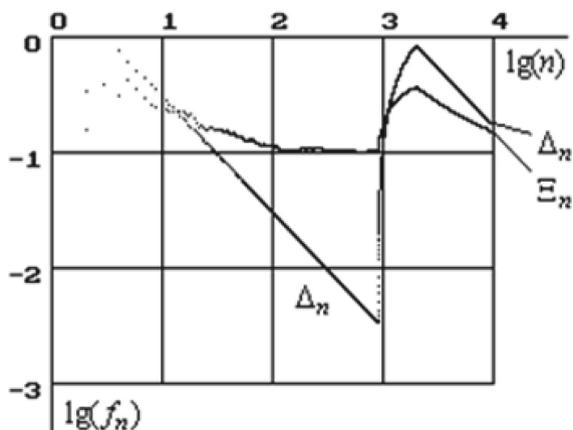


Fig. 6. Method stability illustration

Obtained data is for values $d = 0.01$, and $T = 0.01$ at the moment $n = 1000$, the environment parameter is changed $v_3 = 0.1$ on the weight $v_3 = 1$. This parameter at the moment $n = 2000$ is restored to its original value $v_3 = 0.1$.

Retraining is at the interval steps $n = 1000 \dots 2000$ to search for other optimal strategies according to changing environmental parameters. The initial order of the convergence rate of the method is preserved after resetting the parameter value $v_3 = 0.1$.

6 Conclusions

The adaptive method for forming decision-making agent behaviour strategies in the game with the environment in this work is investigated. The behavioural patterns reproduction based on the stochastic game model reveals the causes and mechanisms for their formation in nature, better help to understand the natural evolution laws and helps to apply game algorithms in practice for various decision support systems constriction in bioinformatics.

The developed decision-making method provides stochastic minimisation of the agent's mean loss function. This method is based on adaptive parametric environment identification using Q -learning and defining a mixed strategy on the Boltzmann distribution. The method convergence is ensured by observing the primary stochastic approximation conditions. The convergence parameters are theoretically determined and refined experimentally. The technique provides

close to 1 order of asymptotic convergence with proper parameter selection. It is the limit value for a given class of recurrent methods.

The proposed method development is possible by considering the state dynamics of the decision-making environment and the transition from a reactive to a cognitive model of the agent.

References

1. Babichev, S., Lytvynenko, V., Osypenko, V.: Implementation of the objective clustering inductive technology based on DBSCAN clustering algorithm. In: Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017, vol. 1, pp. 479–484 (2017). <https://doi.org/10.1109/STC-CSIT.2017.8098832>
2. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopol. Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
3. Bowles, J., Silvina, A.: Model checking cancer automata. In: IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 376–379 (2016). <https://doi.org/10.1109/BHI.2016.7455913>
4. Flieger, S.: Implementing the patient-centered medical home in complex adaptive systems: becoming a relationship-centered patient-centered medical home. *Health Care Manage. Rev.* **42**(2), 112–121 (2017). <https://doi.org/10.1097/HMR.000000000000100>
5. Fricke, G., Letendre, K., Moses, M., Cannon, J.: Persistence and adaptation in immunity: T cells balance the extent and thoroughness of search. *PLoS Comput. Biol.* **12**(3) (2016). <https://doi.org/10.1371/journal.pcbi.1004818>
6. Huang, K., Zheng, X., Cheng, Y., Yang, Y.: Behavior-based cellular automaton model for pedestrian dynamics. *Appl. Math. Comput.* **292**, 417–424 (2017). <https://doi.org/10.1016/j.amc.2016.07.002>
7. Kravets, P., Lytvyn, V., Dobrotvor, I., Sachenko, O., Vysotska, V., Sachenko, A.: Matrix stochastic game with q-learning for multi-agent systems. *Lecture Notes on Data Engineering and Communications Technologies* **83**, 304–314 (2021). https://doi.org/10.1007/978-3-030-80472-5_26
8. Marasanov, V., Stepanchikov, D., Sharko, A., Sharko, O.: Technology for determining the residual life of metal structures under conditions of combined loading according to acoustic emission measurements. *Commun. Comput. Inf. Sci.* **1158** (2020). https://doi.org/10.1007/978-3-030-61656-4_13
9. Mohamed, W., Hamza, A.: Medical image registration using stochastic optimisation. *Opt. Lasers Eng.* **48**(12), 1213–1223 (2010). <https://doi.org/10.1016/j.optlaseng.2010.06.011>
10. Narendra, K.S., Thathachar, M.A.L.: Learning automata - a survey. *IEEE Trans. Syst. Man Cybern.* **4**, 323–334 (1974)
11. Ning, C., You, F.: Data-driven adaptive nested robust optimisation: general modeling framework and efficient computational algorithm for decision making under uncertainty. *AICHE J.* **63**(9), 3790–3817 (2017). <https://doi.org/10.1002/aic.15717>
12. Ozcift, A., Gulden, A.: Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. *Comput. Meth. Program. Biomed.* **104**(3), 443–451 (2011). <https://doi.org/10.1016/j.cmpb.2011.03.018>

13. Robbins, H.: Some aspects of the sequential design of experiments. *Bull. Am. Math. Soc.* **58**(5), 527–535 (1952). <https://doi.org/10.1016/j.ins.2020.06.069>
14. Stewart, A., Bosch, N., D’mello, S.: Generalizability of face-based mind wandering detection across task contexts. In: The 10th International Conference on Educational Data Mining Society, pp. 88–95 (2017)
15. Sutton, R., Barto, A.: Reinforcement learning: an introduction (2017). <http://incompleteideas.net/book/bookdraft2017nov5.pdf>
16. Tsetlin, M.L.: Automaton Theory and Modeling of Biological Systems. Academic Press Inc., New York (1973)
17. Wooldridge, M.: An Introduction to Multi-Agent Systems. John Wiley and Sons, Hoboken, New Jersey, U.S. (2009)
18. Yurtkuran, A., Emel, E.: An adaptive artificial bee colony algorithm for global optimisation. *Appl. Math. Comput.* **271**, 1004–1023 (2015). <https://doi.org/10.1016/j.amc.2015.09.064>



System Analysis of the Internal and External Migration Processes in Ukraine

Andrii Roskladka¹(✉) , Nataliia Roskladka¹ , Olexander Romanyuk² ,
Tetiana Troianovska-Korobeinikova² , and Liudmyla Savytska² 

¹ State University of Economics and Trade, Kyiv, Ukraine
{a.roskladka,n.roskladka}@knute.edu.ua

² Lviv Polytechnic National University, Lviv, Ukraine
tetiana.i.korobeinikova@lpnu.ua, savytska.liudmyla@vntu.edu.ua

Abstract. The object of research is the internal and external migration flows of the Ukrainian people. The solution of the problems put forward in the research, requires scientific systematic approach to the study of both internal and external migration flows, it enables to identify and predict trends and directions of migration flows. The use of the principles and tools of information support to analyze trends in the world market of goods and services, identify critical industries and the ability to involve as many young people as possible to employment in Ukraine are conditions for effective state planning. The study has identified and demonstrated main trends in the internal and external migration. The migration flows of various economically active groups of the population of Ukraine, such as students, workers, etc., are analyzed in the paper. The main ways and causes of migration processes and their influence on the economic and political situation in Ukraine are considered. The promising demographic situation of large cities in Ukraine has been carefully studied. The basic methods of the research are the application of a systematic approach, analysis and synthesis of information and rational use of information technologies to obtain the required results.

The paper presents new visualizations of demographic problems. To demonstrate the trends, research was performed in various spheres and analytical platforms, primarily R and Power BI.

Keywords: Migration flows · Migration processes · Labor migrants · Population forecasting · R programming · Power BI analytical platform

1 Introduction

Modern migration processes in Ukraine are closely connected with people who want to live in better conditions and receive a higher salary. In the past, the fate of a state, region or city was determined primarily by the availability of natural resources. Everything has changed dramatically now. Countries rich in minerals

and resources are not necessarily countries with high wages. Human capital is the main asset of the developed countries.

Ukraine is a co-initiator and active participant in most international legal institutions that regulate the rights of migrants. These are the “Universal Declaration of Human Rights”, the “European Convention on Human Rights”, the “Geneva Convention relating to the Status of Refugees”, numerous documents against the smuggling of migrants and human trafficking, which are part of the “UN Convention against Transnational Organized Crime”.

However, compliance with international conventions does not guarantee optimal management of migration flows. Population migration has long been a global problem. The process of globalization is constantly increasing the number of citizens living outside their home country. According to the United Nations, there are more than 244 million migrants now, which is 40% more than in 2000. 232 million people a year become international migrants and another 740 million within the country. The population of Ukraine is steadily decreasing, owing also to emigration from the country, which negatively affects many internal processes.

The war of the Russian Federation against Ukraine has significantly aggravated the migration problems. According to the UN, war uprooted more than 12 million Ukrainians (more than 5 million Ukrainians went abroad, and another 7.1 million are internally displaced persons who remained in Ukraine). Migration processes are natural, but only if the approximate balance between migrants is maintained. A significant imbalance towards emigration or immigration always negatively affects social-economic development of the country. For Ukraine, this means that the state should have a clear plan on how to stop the outflow of migrants.

The elaboration of such plan will be facilitated by the results of the authors’ research presented in this paper. When covering migration problems, only the total number of migrants is often presented. To identify trends in the internal and external migration processes, the authors study in detail certain categories of migrants. System analysis of migration processes, as well as the use of modern tools of data analysis, made it possible to create a qualitative forecast of migration processes in Ukraine for the coming decades.

The main contributions of this paper are summarized as follows.

1. The state of migration processes is investigated and the basic principles of migration policy of Ukraine are analyzed.
2. Problems of the number of migrants accounting and ways to overcome them are identified.
3. The analysis of internal migration of Ukrainians within the country is carried out. The main directions of internal migration flows are identified. The trend to the excessive urbanization of the population is observed. A forecast of the population size in the largest cities of Ukraine until 2050 is made.
4. The external migration of Ukrainians is analyzed. Some categories of the external migrants, primarily migrant workers and students, are studied. Groups of external labor migrants are divided according to regional

preferences in relation to foreign countries. Statistical analysis of incoming and outgoing student flows is conducted.

5. Forecasts for different categories of migrants are made by means of R statistical analysis environment and the Power BI analytical platform.

The rest of the paper is structured in a following way.

Sect. 2 provides a review of the literature, containing the results of various studies of migration flows. The lack of a systematic approach to the analysis of migration flows is noted.

Section 3 contains the problem statement and the description of the main reasons that significantly complicate the analysis of migration flows.

Section 4 presents the results of statistical analysis in R and strategic analysis of different categories of Ukrainian migrants.

Section 5 provides a general analysis of the incoming and outgoing migration flows from Ukraine, a forecast and visualization of the forecast models of migration flows in Power BI is made.

The “Conclusions” section summarizes the results of the study and provides some recommendations for improving the migration policy of Ukraine.

2 Literature Review

Fundamental rights in the field of migration are enshrined in many normative acts of Ukraine, including the laws “On Citizenship of Ukraine”, “On Freedom of Movement and Free Choice of Residence”, “On the Procedure for Leaving Ukraine and Entering Ukraine by Citizens of Ukraine”, “On External Labor Migration”, “On the Legal Status of Foreigners and Stateless Persons”, “On Immigration”, “On Refugees and Persons in Need of Additional or Temporary Protection”, “On Ukrainians Abroad”, “On Ensuring the Rights and Freedoms of Internally Displaced Persons”, etc. Main strategic document is the “Strategy of the state migration policy of Ukraine for the period up to 2025” [12], which formulates the main goals of the migration policy of the state and ways to achieve them.

The issue of migration of the country’s population is the subject of research by a large number of specialists in various fields. For example, Irene Mosca and Robert E. Wright in their paper “National and international graduate migration flows” explore the nature of national and international graduate migration flows in the UK [20]. Guy J. Abel studies the problems of estimating the values of migration flows [13]. British researchers Yu-wang Chen, Lei Ni, Luis Ospina-Forero study the impact of internal migration on population structure, utilities, economic and social development of local areas [16]. In this paper, the authors use modern tools Visual Mining to visualize the results of internal migration study. Progressive approaches to the prediction of refugee migration processes are being investigated by M. Carammia, S. M. Iacus and T. Wilkin in [15].

The European Commission constantly organizes competitions of research projects on migration processes [2].

Recently, much attention has been paid to modern information tools for the research of migration flows. The authors believe that the most promising environment in the construction of predictive models in the analysis of big data is the system of statistical modeling R [26–29]. The Microsoft Power BI analytical platform should be used to visualize of the results of analytical research [21, 28].

The authors of the paper have considerable experience in studying the migration issues, in particular, in applying mathematical and computer modeling methods for research [19, 22–25].

Thus, most authors study individual categories of migrants and the corresponding directions of migration flows. For the elaboration of the effective plan to combat excessive migration, a systematic analysis of migration processes based on modern data analysis technologies and advanced software for business analytics is necessary. This is the only way to hope for a high-quality analytical forecast of migration processes and an adequate migration policy of the state.

3 Problem Statement

Problems, caused by the spread of the coronavirus, the imposition of martial law as a result of the Russian Federation's full-scale aggression against Ukraine, significant restrictions on the movement of citizens within the country and almost complete closure of borders made it impossible to use classical tools for long-term economic forecasting and assessment of development potential [17, 18] and have further complicated the study of migration flows in Ukraine. Solving these issues requires radical economic transformation and the use of innovative tools to predict trends in the internal and external migration by means of mathematical and computer modeling to manage migration flows at the regional and national levels.

Every decision on migration policy requires reliable information support based on the calculation and analysis of quantitative indicators based on statistical data [10]. Despite the declaration on the availability of statistical data in the field of population migration, the national statistical system is characterized by a number of internal inconsistencies, which significantly complicates the further use of data for strategic decision-making [19]:

- time lag of information flows from the market needs. Part of the data for 2020 and 2021 remained unpublished at the time of writing;
- lack of time series for some data groups;
- loss of a significant part of the internal migration flows due to limitations of relevant statistical observations;
- inconsistency of data, obtained from different sources;
- an obvious inaccuracy of some administrative data due to the imperfection of the relevant procedures for obtaining them;
- the inconvenience of presenting and searching of information that is currently presented on different platforms and in extremely inconsistent forms.

However, these issues should not force us to abandon the study and prediction of migration flows, but only encourage researchers to implement systematic scientific approaches to the study of this important problem.

The aim of the paper is systematic approach to studying, modeling and forecasting of the internal and external migration flows.

4 Materials and Methods

The subject of research is models and information technologies for the analysis of migration flows. The basic methods of the research are the application of the systematic approach, analysis and synthesis of information and rational use of information technologies to obtain the required results.

In particular, the study uses the following scientific methods:

- method of system analysis – to determine the most important and influential indicators of migration flows in the regions of Ukraine and foreign countries;
- graphical method – to create interactive charts in R environment (ggplot2 packages – to create static charts, gganimate – to create animated charts, tidyverse – to manage data migration flows, janitor – to clear data arrays, scales – to format created charts);
- forecasting methods – to create a forecasting model in the R environment for the internal and external migration, to create a forecast in the analytical platform Power BI in order to determine trends in population inflows and outflows in the years to come;
- method of quantitative analysis – to study the structure of migration flows, determine their statistical characteristics for different types of economically active population.

The information base of the study is statistical data on the volume of migration flows provided in the regions of Ukraine, data from the State Statistics Service of Ukraine [8], State Migration Service of Ukraine [7], State Border Guard Service of Ukraine [6], official foreign statistical sources and other materials, collected by the authors of the paper, in particular, during the formation of the Tourism Barometer of Ukraine [10].

4.1 Statistical Analysis of the Population in Ukraine

Ukraine has an area of 603.5 thousand square kilometers and it is the largest country, which lies entirely within Europe. As of 2020, the population of Ukraine is 41.9 million people, excluding the annexed territory of Crimea. The gender composition of the population of Ukraine is as follows: 46.3% – male, 53.7% – female. 71.1% live in urban areas and only 28.9% – in rural areas. Life expectancy at birth is 71.76 years (66.89 years for men and 76.72 years for women). The average age of Ukrainians is 40.6 years, and the country is facing a stable aging of the nation.

According to the State Statistics Service of Ukraine, every year more than half a million of Ukrainians change their place of residence. However, this figure is based on information from the registration authorities and covers only those relocations where persons have registered a change of place of residence.

At the same time, according to sociologists, at least 12% of adult Ukrainians [9], and according to some estimates in large cities from 15 to 30%, do not live in the place where they are registered. Thus, internal migration is much more active, than it can be seen from statistics. It is primarily focused on cities. Traditionally, the capital Kyiv has a positive migration balance, as well as the regions where the largest cities of Ukraine are located – Dnipro, Kharkiv, Odesa, Lviv.

Figure 1 shows fragments of an animated diagram of population residence change in major Ukrainian cities. The forecast values are calculated on the basis of international prospects for urbanization growth (from 67.2% in 2001 and 71.1% in 2019 to 72–73% in 2030) and overall population decline [4]. The population of Ukraine is steadily decreasing from its peak in 1992, when the population was about 52 million, to the current 41.9 million, without taking into account the population of Crimea. According to forecasts, over the next 30 years Ukraine may lose another 7.05 million people or up to 15.8% of the population.

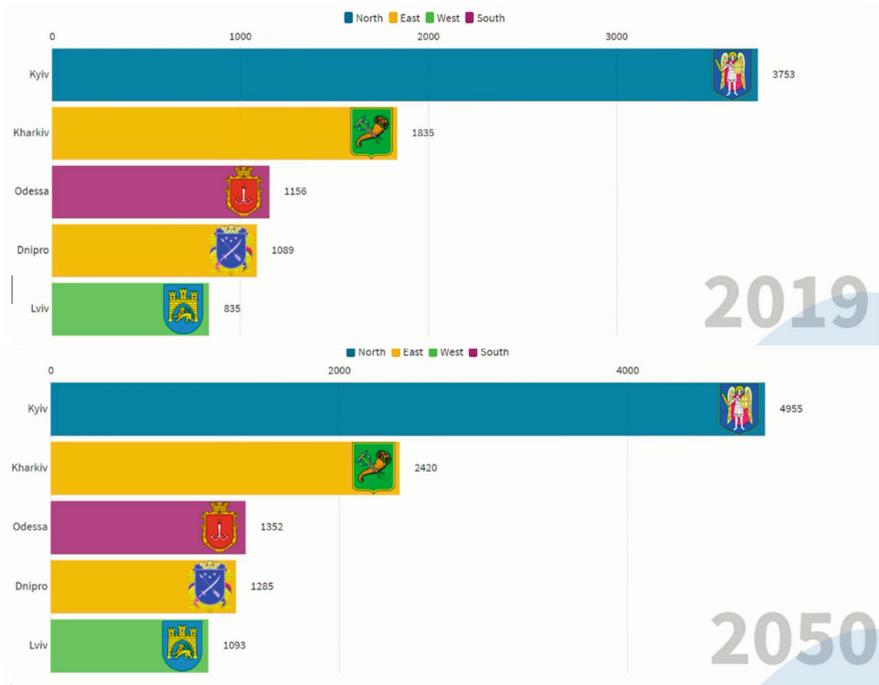


Fig. 1. The number of residents of major cities of Ukraine and the forecast until 2050

4.2 Analysis of Internal Migration in Ukraine

After gaining independence, the volume of the internal migration in Ukraine significantly decreased due to the disappearance of organized resettlement and recruitment, commercialization of education, housing prices increase, employment difficulties, etc. According to the State Statistics Service, in 2015 only 12 out of a thousand Ukrainians changed their place of residence [8]. However, domestic migration, which is recorded statistically, is about 10 times larger than international migration.

In interregional migrations, the main recipients are Kyiv and Kyiv region. The participation of the population in internal labor migrations, that are not related to daily or weekly return to the place of permanent residence, is high. Such migration is typical for seasonal, shift and longer work in the capital and major industrial centers.

According to the latest data, obtained during the study conducted by the International Organization for Migration in 2015–2019, the number of internal labor migrants in Ukraine exceeds 1.7 million and annually reaches 9% of the economically active population. Moreover, in the near future, internal labor migration may increase by about 50% [5].

In Fig. 2, main migration flows within Ukraine are identified: Kryvyi Rih-Dnipro-Zaporizhzhia agglomeration, Odesa agglomeration, Kyiv region, Kharkiv and Lviv. The predicted situation with the internal migration has begun to change dramatically since 2014, when Ukraine is experiencing an incessant armed conflict in the densely populated eastern region, as a result of this conflict there appeared people with a clear status of the internally displaced persons. The annexation of the Crimea and the conflict in the eastern Ukraine have led to numerous forced displacements, especially in the early years of the conflict. As of the end of 2021, more than 1.42 million internally displaced persons were registered in social welfare bodies.

Figure 3 presents the fragment of an animated chart with data, based on OCSE reports and reports of the Ukrainian Migration Service on the most active internal migration from June 2014 to November 2019 [1]. Three regions with the highest number of the internally displaced persons are Donetsk, Luhansk regions and city of Kyiv. Many internally displaced persons are also in Kharkiv, Dnipropetrovsk and Zaporizhzhia regions. The number of internally displaced persons reached its highest level in the first half of 2016, and then it decreased. At that time, the hostilities were suspended, and people began to feel safe in some regions of Donbas.

At the same time, the surveys show that the share of the internal migrants who do not intend to return to their places of residence even after the end of the conflict, constantly grows. In June 2019, the number of the internal migrants was 34% more as compared with October 2017, when approximately 26% of respondents said they did not want to return to certain regions of Donetsk and Luhansk regions.

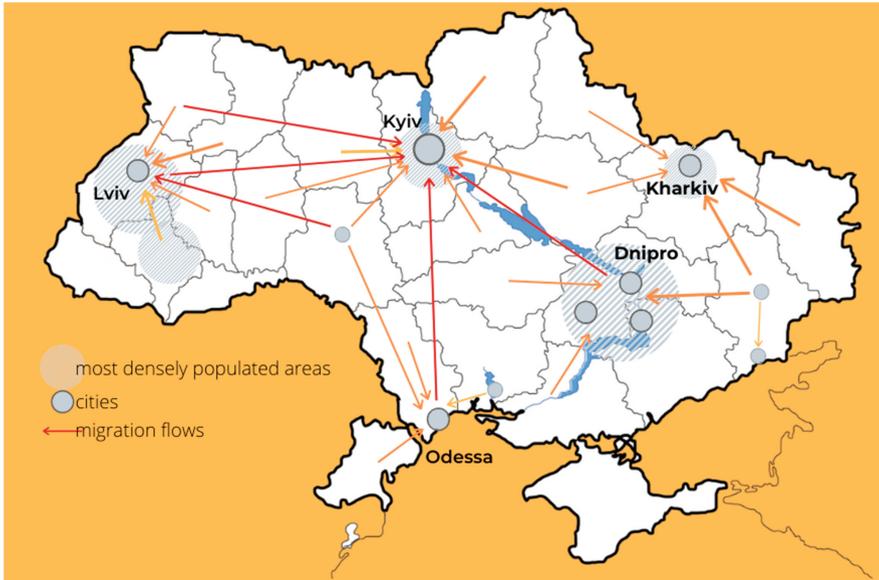


Fig. 2. Main directions of internal flows in Ukraine

On February 24, 2022, as a result of the aggression of the Russian Federation, internal migration in Ukraine became huge. Despite numerous assessments of internally displaced persons, official data should be expected only after Ukraine's victory in the war. The authors plan to conduct a separate large-scale study of this problem.

4.3 Analysis of External Migration in Ukraine

Figure 4 shows the fragment of the animation regarding the percentage of Ukrainian labor migrants abroad in countries, based on data from the International Organization for Migration from 2010 to 2019 [5,8].

Let us to analyze main places of employment and interest of Ukrainians abroad. Based on the conducted research, two maps are constructed. The first map is for 2010–2012 (Fig. 5), the second is for 2015–2019 (Fig. 6) [14].

Depending on the color of the region, the disposition of the residents of this region to the external labor migration is determined. In 2010–2012, we identified 5 types of regions according to the number of international migrants. The lowest indices (regions where people are the least interested in working abroad) possess such cities as Kyiv, Odesa, Dnipropetrovsk, Poltava and Kyiv regions. The largest number of labor migrants is observed in Zakarpattia and Bukovyna. In addition, high level of seasonal, temporary and permanent migrants is observed in Western Ukraine.

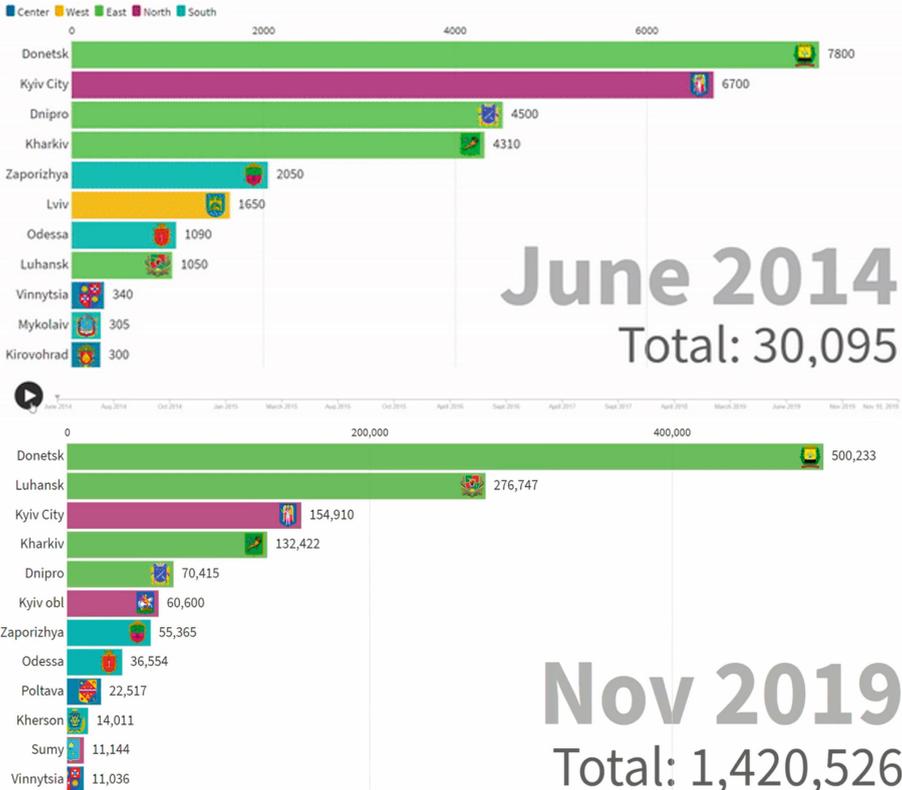


Fig. 3. Number of internally displaced persons in Ukraine, 2014–2019

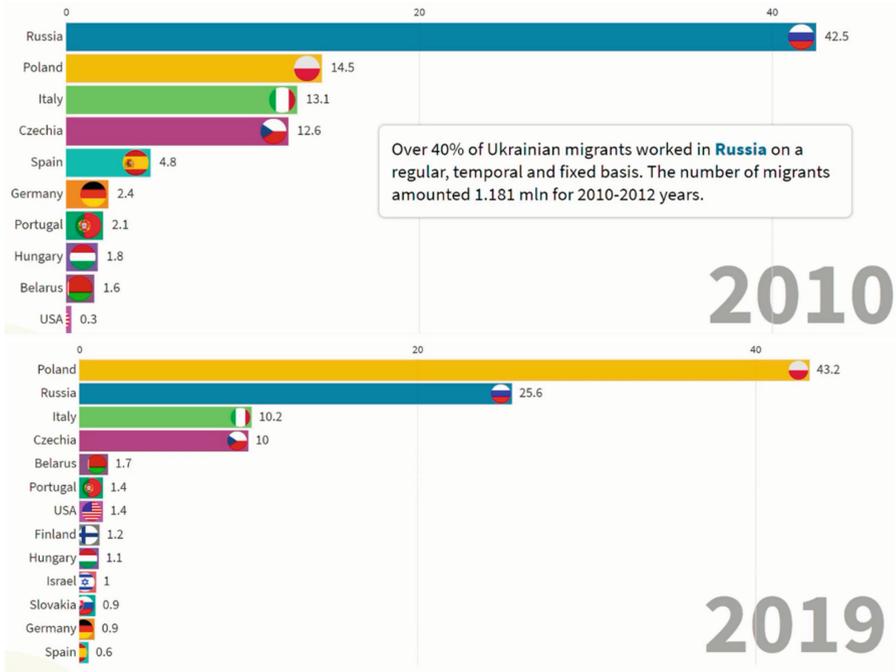


Fig. 4. Percentage of Ukrainian labor migrants abroad

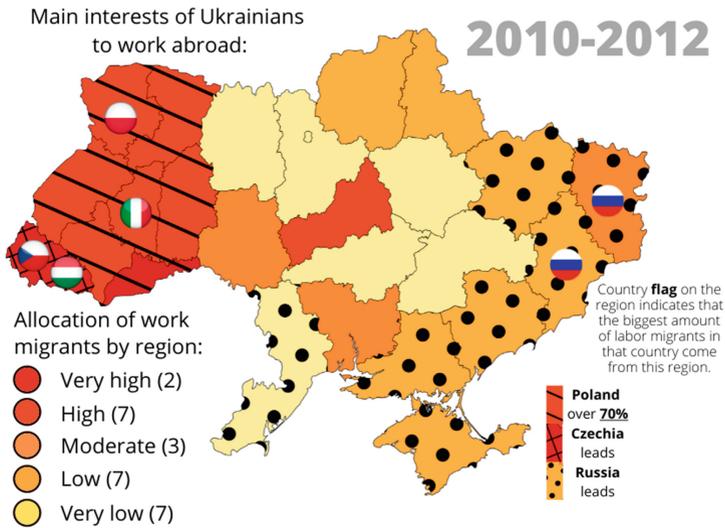


Fig. 5. Main directions of work and interests of Ukrainians abroad (2010–2012)

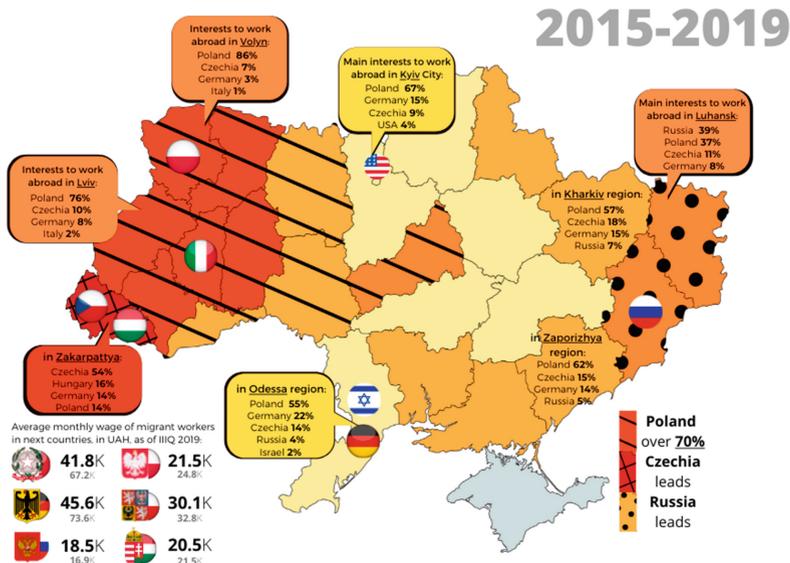


Fig. 6. Main areas of work and interests of Ukrainians abroad (2015–2019)

In the regions of Western Ukraine labor migrants prefer Poland and in Zakarpattia they prefer the Czech Republic and Hungary. Italy prevails as a country for migrants in the Ternopil region, and Russia are traditionally preferred in Eastern regions.

In the period 2015–2019, the distribution of labor migrants changed slightly (Fig. 6). Poland is the most common choice for people, working abroad. In 10 regions of Ukraine, more than 70% of external labor migrant work in Poland. Before the war with Ukraine, Russia continued to be attractive for two regions in the East.

Significant expansion of the geography of countries that Ukrainians choose to work in may be observed. The indices of Israel, Finland, the United States and other countries have increased as compared with 2010–2012.

Main factor forcing migrant workers to look for the work abroad is the wage rate. Currently, the average wage in Poland is twice as high as in Ukraine. Germany and Italy offer wages that are almost four times higher than in Ukraine, but according to the Ukrainian Migration Service, less than 11% of migrant workers prefer the German and Italian markets.

According to UNESCO, the number of Ukrainian students abroad is growing steadily, and after 2013 it almost doubled to 77.6 thousand (Fig. 7). With the stabilization of the situation in the country, this process has slowed down. As of 2019, the number of Ukrainian students abroad has decreased to 73 thousand people [3]. The situation in 2022 is poorly predicted even for professionals in the field of education services.

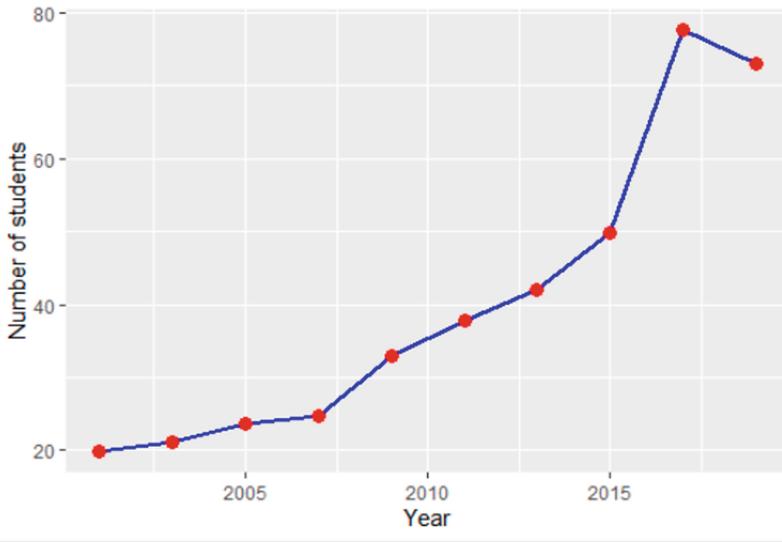


Fig. 7. The number of Ukrainian students studying in foreign free economic zones (thousand people)

The most popular countries where Ukrainian students go to get higher education [3]:

1. Poland is over 45% (share has doubled as compared with 2014)
2. Russia is 19% (share has been steadily decreasing since 2014)
3. Germany is more than 8% (often as a choice for a master's degree)
4. Czech Republic is 6.6% (the share has been growing slightly in recent years)

The vast majority of Ukrainian students go to study in the economic and geographical region of Europe and North America.

The desire to study abroad is often associated with the desire of young people to find their way to the international labor market. Thus, out of more than a thousand Ukrainians studying in Poland, only 6% of them intended to return to Ukraine after graduation.

The growth of educational migration is also indirectly confirmed by the fact that the families of Ukrainian labor migrants are being reunited in their countries of residence. This is probably due to the rapid growth of the number of Ukrainian students in countries where labor migration has led to numerous Ukrainian diasporas. For example, in 2000 only 40 Ukrainians studied at Italian Universities, and in 2017 - 2.8 thousand, which is 70 times more.

On the positive side, only 5% of Ukrainian applicants who have passed external tests with 180 or more points leave Ukraine and go to study in the European Union.

Our country also accepts students from other countries. People from the economic and geographical regions of Central and South Asia and Africa mostly come to Ukraine to study.

According to the latest UNESCO reports, there are more than 55,000 foreign students in Ukraine. This number has been actively increasing for 10 years. The most attractive cities for foreign students are Kharkiv (20 thousand foreigners studied in 2021), Kyiv (14 thousand foreign students), Odesa (7.5 thousand), Dnipro (3.7 thousand) and Zaporizhzhia (3 thousand students from other countries) [11].

The most popular countries from which higher education students go to Ukraine:

1. India is 19.3% (more than 10.7 thousand people)
2. Azerbaijan is 10% (5.5 thousand students)
3. Morocco is 8.4% (4.7 thousand, the highest growth rate)
4. Turkmenistan is 6.7% (3.8 thousand)
5. Nigeria is 5% (2.8 thousand students)
6. Israel is 3.6% (2 thousand)
7. Poland is 1.3% (about 700 students)
8. United States is 0.5% (about 300 students)

Note that the share of students from our most attractive regions for educational migration from Europe and North America is only 4.5%.

5 Results and Discussions

The authors of the paper are one of the authors of the “Tourism Barometer of Ukraine” - an innovative statistical publication, which was presented in 2020 by the National Tourism Organization of Ukraine [10]. Clear and accessible tourism statistics becomes the basis for strategic decisions of both entrepreneurs, state and local authorities, including the migration policy of Ukraine.

Thanks to significant team efforts, the authors managed to create a system of interactive dashboards in the field of tourism with actual data at the time of its development. Figure 8 presents a dashboard, which is created in the analytical platform Power BI. It defines quantitative and graphical characteristics of external migration flows. The left part of the dashboard contains the comparative description of the countries where Ukrainian citizens traveled during last 10 years. The right side of the dashboard contains a graphic description of foreign citizens who came to Ukraine from different countries during last 10 years.

Indicative are flows of the internal migration, developed in Power BI and presented in Figs. 9, 10.

Figure 9 contains information on collective accommodation facilities (CAF) by the regions of Ukraine, which are one of the main sources of information about the internally displaced persons.

Figure 10 gives an opportunity to conduct a comparative analysis of the number of Ukrainian and foreign tourists in the regions of Ukraine who stayed in the CAF for the last 10 years. Dashboard also reveals positive or negative growth rates of the internal and external migrants.

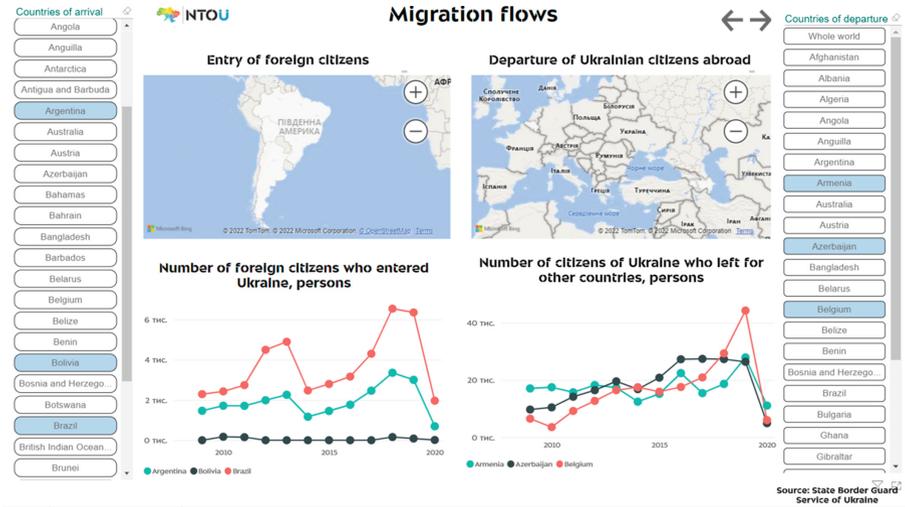


Fig. 8. Dashboard with incoming and outgoing migration flows from Ukraine

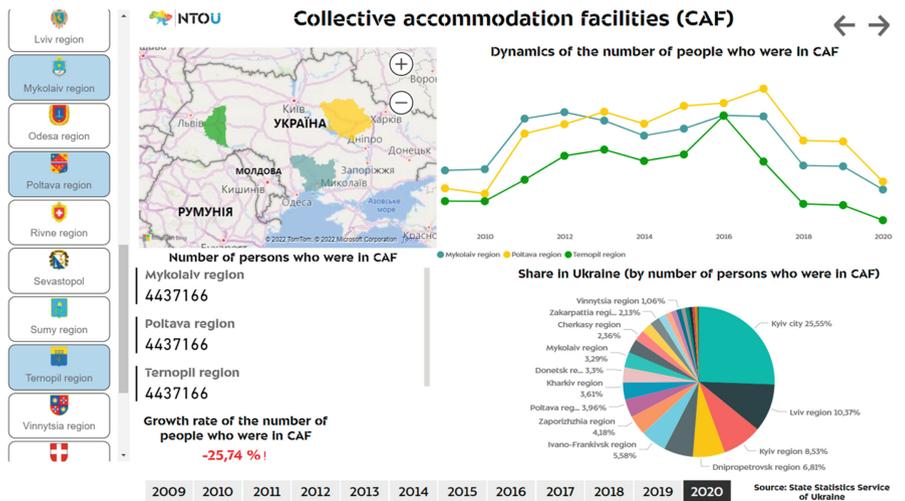


Fig. 9. Interactive visualization of collective accommodation facilities in Ukraine

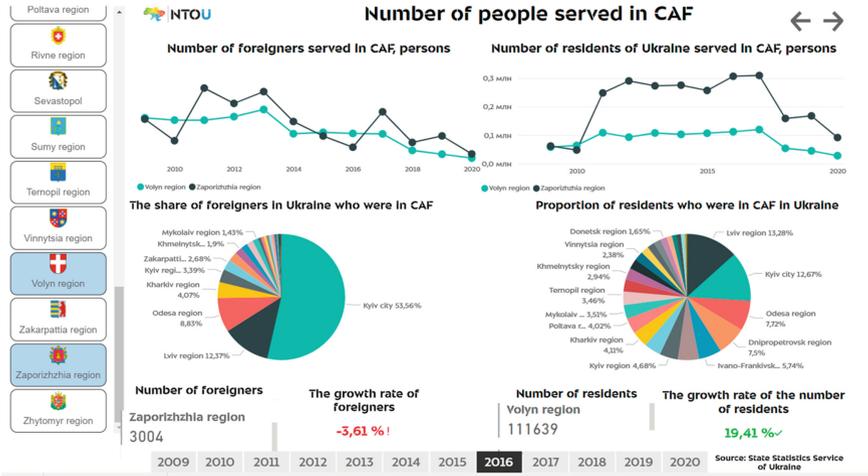


Fig. 10. Interactive visualization of the number of Ukrainian and foreign tourists by regions of Ukraine

In addition to the efficient interactive visualizations, Power BI also comprises tools for data processing, including the construction of the predictive models. Forecasting in Power BI is based on the method of exponential smoothing for time series prediction. The exponential smoothing method showed good results in both scientific and business applications and has the advantage that it suppresses noise or unwanted variations, which can distort the model, at the same time the given method efficiently captures the trends.

Figure 11 presents the implementation of the forecast models, that makes it possible to determine the estimated total number of people who will arrive in Ukraine and go abroad in 2023.

It should be noted, that the current difficult military situation in Ukraine will significantly affect the indices of the migration flows. However, there are no official data on the flow of the internal and external migrants for the current year.

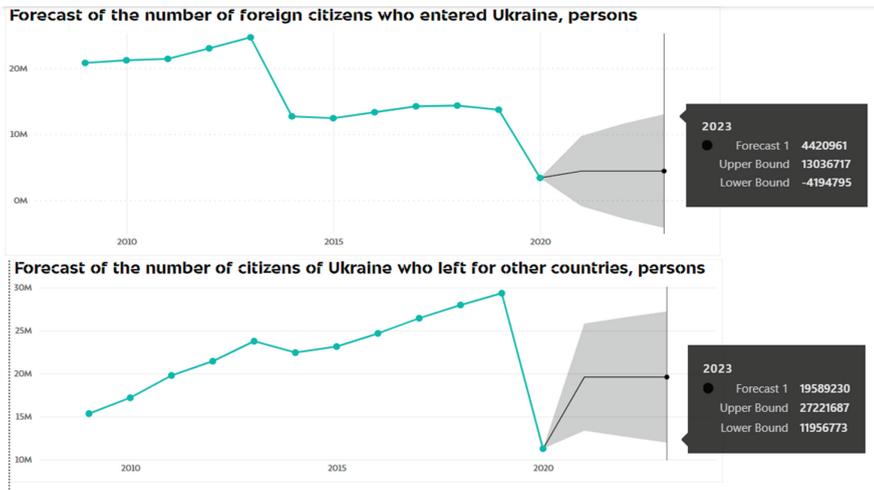


Fig. 11. Visualization of predictive models of the migration flows in Power BI

6 Conclusions

At the moment, we witness the demographic crisis in Ukraine, that depends primarily on the level of living conditions and social welfare. The demographic crisis started more than 30 years ago and aggravates with each passing year.

The projected situation with the internal migration has begun to change dramatically since 2014, when Ukraine is experiencing an incessant armed conflict in the densely populated eastern region, as a result of this conflict there appeared people with a clear status of the internally displaced persons. The annexation of the Crimea and the conflict in eastern Ukraine have led to numerous forced displacements, especially in the first years of the conflict.

As a result of the current aggression of the Russian Federation internal and external migration in Ukraine has become huge and practically uncontrolled.

To solve this problem, the scientific systematic approach to the study of the internal and external migration flows is necessary, which will identify and predict trends and directions of migration flows. The use of the principles and tools of information support to analyze trends in the world market of goods and services, identify critical industries and the ability to involve as many young people as possible to employment in Ukraine are conditions for effective state planning.

The study has identified and demonstrated the main trends both within the country and for external migration. The paper analyzes the migration flows of various economically active groups of the population of Ukraine, including students, employees and others.

To demonstrate the trends, the research was carried out in various environments and analytical platforms, primarily R and Power BI.

References

1. Daily and spot reports from the special monitoring mission to Ukraine. <https://www.osce.org/ukraine-smm/reports>
2. European commission research projects. migration and mobility. <https://ec.europa.eu/info/research-and-innovation/research-area/social-sciences-and-humanities/migration-and-mobility>
3. Global flow of tertiary-level students. <http://uis.unesco.org/en/uis-student-flow>
4. The global talent competitiveness index. <https://gtcistudy.com/the-gtci-index/>
5. Migration as a factor of development in Ukraine. <https://ukraine.iom.int>
6. State border guard service of Ukraine. <https://dpsu.gov.ua>
7. State migration service of Ukraine. <https://dmsu.gov.ua/diyalnist/statistichni-dani/statistika-z-osnovnoj-diyalnosti.html>
8. State statistics service of Ukraine. <https://ukrstat.org/en>
9. The system of registration of residence in Ukraine: its impact on the realization of human rights and vital interests. <https://cedos.org.ua/researches/>
10. Tourism barometer of Ukraine. <http://www.ntoukraine.org/assets/files/ntou-barometer-2020.pdf>
11. We will show the world that Ukrainian universities are changing for the better. <https://www.radiosvoboda.org/a/27813828.html>
12. On approval of the strategy of the state migration policy of Ukraine for the period up to 2025 (2017). <https://zakon.rada.gov.ua/laws/main/482-2017-%D1%80>
13. Abel, G.J.: Estimates of global bilateral migration flows by gender between 1960 and 2015. *Int. Migrat. Rev.* **52**(3), 809–852 (2020). <https://doi.org/10.1111/imre.12327>
14. Baiev, R., Roskladka, A.: Simulation of migration flows in Ukraine. In: *Digital Economy: Collection of Scientific Papers of Students*, pp. 10–14 (2020)
15. Carammia, M., Iacus, S.M., Wilkin, T.: Forecasting asylum-related migration flows with machine learning and data at scale. *Sci. Rep.* **12**, 1457 (2022). <https://doi.org/10.1038/s41598-022-05241-8>
16. Chen, Y., Ni, L., Ospina-Forero, L.: Visualising internal migration flows across local authorities in England and Wales. *Environ. Plan. Econ. Space* **53**(4), 616–618 (2021). <https://doi.org/10.1177/0308518X20968568>
17. Geseleva, N.V.: Mechanisms of modernization and technological development of Ukraine's economy. *Actual Prob. Econ.* **125**(3), 64–72 (2011)
18. Geseleva, N.V.: Tools for economic growth regulation on the basis of macroeconomic equilibrium. *Actual Prob. Econ.* **143**(5), 32–37 (2013)
19. Kharlamova, G., Roskladka, A., Roskladka, N., Stavyt'skyy, A., Zabaldina, Y.: Cluster analysis of Ukrainian regions regarding the level of investment attractiveness in tourism. In: *Proceedings of the 17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. II*, pp. 401–416 (2021). <http://icteri.org/icteri-2021/proceedings/volume-2/202110401.pdf>
20. Mosca, I., Wright, R.: National and international graduate migration flows. *Popul. Trends* **141**, 36–53 (2010). <https://doi.org/10.1057/pt.2010.20>
21. Powell, B.: *Microsoft Power BI Cookbook: Creating Business Intelligence Solutions of Analytical Data Models, Reports, and Dashboards*, p. 802. Packt Publishing (2017)
22. Roskladka, A., Roskladka, N., Dluhopolskyi, O., Kharlamova, G., Kiziloglu, M.: Data analysis and forecasting of tourism development in Ukraine. *Innov. Mark.* **14**(4), 19–33 (2018). [https://doi.org/10.21511/im.14\(4\).2018.02](https://doi.org/10.21511/im.14(4).2018.02)

23. Roskladka, A., Roskladka, N., Kharlamova, G., Karpuk, A., Stavytskyy, A.: The data science tools for research of emigration processes in Ukraine. *Prob. Perspect. Manag.* **18**(1), 70–81 (2020). [https://doi.org/10.21511/ppm.18\(1\).2020.07](https://doi.org/10.21511/ppm.18(1).2020.07)
24. Roskladka, A., Roskladka, N., Kharlamova, G., Stavytskyy, A.: The data association algorithm for the formation of optional it-courses list system. In: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*, vol. 2732, pp. 515–531 (2020). <http://ceur-ws.org/Vol-2732/20200515.pdf>
25. Roskladka, N., Roskladka, A.: Ukraine and the World: the tourism system, chap. Computer modeling of tourism flows in Ukraine, pp. 41–55. Eastern European Center of the Fundamental Researchers (EECFR) (2019)
26. Shmueli, G., Bruce, P., Yahav, I., Patel, N., Lichtendahl, K.: *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*, p. 544. Hoboken, Willey (2018)
27. Torgo, L.: *Data Mining with R: Learning with Case Studies*, p. 446. Chapman and Hall, London (2018)
28. Wade, R.: *Advanced Analytics in Power BI with R and Python: Ingesting, Transforming, Visualizing*, p. 440. Apress, New York (2020)
29. Wickham, H., Grolemund, G.: *R for Data Science*, p. 522. Newton, O'Reilly Media (2017)



Associative Information Retrieval in Medical Databases

Anatoliy Povorozyuk¹, Anna Filatova¹, Oksana Povorozyuk¹,
and Iryna Shakhina²

¹ National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine

ai.povoroznjuk@gmail.com, povoks@i.ua

² Vinnytsia Mykhailo Kotsiubynskyi State Pedagogical University,
Vinnytsia, Ukraine

Abstract. The purpose of this study is to formalize the problem and develop a method for associative information retrieval in medical databases during the design of computer decision support systems in medicine. A method of associative information retrieval has been developed, based on determining the degree of similarity of text strings by calculating the Levenshtein distance. Similarity criteria between the entered words and words in the database have been developed, which take into account the peculiarities of the Russian language and provide for the possibility of typical errors during entering text from the keyboard. Based on expert assessments of typical keyboard text input errors (case mismatch; erroneous input of characters located nearby on the keyboard, ordinal mismatch of consecutive characters; spelling errors during entering hard and soft vowels, voiced and unvoiced consonants), additional criteria for detection of similarity of 2 characters during calculating the Levenshtein distance have been developed. The method is implemented in a medical decision support system for prescribing drugs in dermatology. Associative search is performed during entering data from the keyboard and forming a request to the “Diagnosis” text field. A test check of the method was carried out, as a result of which various types of distortions of the query word, which may occur during entering this word from the keyboard were performed, and the results of associative search in the database are given. Test verification confirmed the efficiency and effectiveness of the developed method.

Keywords: Databases · Associative retrieval · Text string · Levenshtein distance · Similarity criterion · Peculiarities of the Russian language · Errors when typing text · Medical system

1 Introduction and Literature Review

The availability of information, the volume, and speed of its processing are decisive factors for the development of the productive forces of science, culture, community institutions, etc. Information retrieval systems (IPS) of the medical

profile play an important role [26,27]. IPS perform storage (possibly in a distributed form as part of local or global computer networks), and the information retrieval on specified queries (retrieval descriptors). Information can be stored both in the form of a relational database and in the form of individual hypertext documents.

Access to data from the database is carried out by implementing a query that must be formulated by the user. The DBMS translates this query into a data manipulation language that is convenient for its work, which allows you to perform all the necessary actions on the data: search, replace, etc. SQL is most often used as such a language in modern DBMS [6].

At the same time, a huge disadvantage of existing DBMSs is their formality and rigidity with regard to data entry: the data must be specified in a strictly defined format and exactly match the information existing in the database, otherwise the search will not be successful. And a user, especially unqualified, may make an error (or errors) when form a request and the implementation of such a request using standard data retrieval tools will be impossible. Therefore, the DBMS must have an intelligent interface that would allow such errors to be detected.

In particular, this problem is relevant when working with text fields of DB. The only means of comparing two text strings in SQL is the LIKE function. It analyzes text strings, and at the same time, those that differ by one, two letters at the end of a word are considered similar. Thus, if in the request only the case of the first letter is changed then the LIKE function will not find the similarity of such strings.

The task of detecting the similarity of text strings is also relevant for the retrieval for documents in global networks (in particular, the Internet), where the retrieval is performed by keywords [22]. Text information (authors and titles of books, titles of magazines) is used as the retrieval keywords. In this case, successful completion of the retrieval is guaranteed with an error-free request. But the user does not always know the necessary information for the implementation of such requests. Therefore, contextual information retrieval is more often used (by keywords or terms that are present in the required text documents) [5,11]. The contextual query is fuzzy because keywords may be in a document in various phrases and cases [13,16,23].

To implement contextual retrieval, text documents are stored as hypertext, that is, a document is represented as a set of descriptors (keywords) linked by reciprocal links, and an explanatory block of text information is attached to each descriptor. Converting plain text into hypertext is called indexing of a text document and it is, in fact, the construction of a retrieval image of a document.

Documentary information retrieval systems are divided into two classes: systems with Boolean and vector retrieval [18]. In Boolean retrieval, only those documents are issued that exactly correspond to the formulated query. In systems with vector retrieval, a query in natural language is used as a request, for each document a measure of its compliance with the query is calculated, documents are issued in a ranked form in descending order of their compliance with

the query. Thus, these systems also require an intelligent interface that allows performing associative (vector) retrieval, calculating the degree of similarity of text strings that represent the query and retrieval image of the document.

Let us consider methods for implementing associative queries when searching the database. The complex problem of searching for documents in textual databases is considered in [1], where the most promising solution to this problem is the use of the most similarity methods.

One of the popular methods for implementing associative queries is an approach based on semantic analysis of queries in natural language [12, 25]. The database retrieval is performed based on the correspondence between the semantic structures of the user query and the stored document.

Methods and software for selecting thematic textual information are considered in [2]. Retrieval by keys is combined with the semantic analysis of natural language texts for the selection of relevant information. It is believed that the most important user information is contained in the core constructions of sentences which consist of the subject, the predicate, and the other members of the sentence connecting them. At the same time, the cyclical formation of keywords is carried out in such a way as to fully disclose the topic under study through the definitions and connections of terms.

In [30], an analysis of a natural language text and its formal representation are performed which allows it to be correctly interpreted and processed. Different types of links in phrases are considered. The article discusses the use of formal grammar to describe the syntax of phrases of the Russian language and the analysis of phrases using the technology of LR-analysis.

To solve problems of morphological and syntactic analysis of the text, as well as problems of inflection analysis, a semantic neural network is used [14, 15, 28], the number of layers of which corresponds to the number of analysis levels (morphological, syntactic and semantic), and each level is implemented as a binary synchronized linear tree with a timeline. Along with the semantic analysis of information, various methods of error detection are used when entering symbolic information [8, 9, 29]. For example, the system SLIPS/1 is developed in [29] for simulating phonemic errors. Analysis and classification of the most common phonological errors are carried out.

The next approach in the implementation of an associative query is an approach in which the query itself and key fields or their combination from the database are represented by points in a multidimensional feature space, and the implementation of an associative query is reduced to the task of calculating some similarity measure between these points [3]. The implementation of associative retrieval based on Kohonen self-organizing maps (SOM), which have the favorable property of preserving topology, is considered in [7, 17, 20].

From the above review, it is clear that standard means for implementing queries is not enough of implementing an associative retrieval in the database, and each of the considered methods of associative retrieval has its advantages and disadvantages, as well as its scope. Therefore, the actual task is to develop new methods of associative retrieval, in particular, using various methods for

detecting the similarity of text strings when designing medical decision support systems with an intelligent interface [19,21,32].

2 Problem Statement

The aim of the work is to develop a method of associative information retrieval and its implementation when designing an intelligent interface in medical decision support systems, which takes into account the peculiarities of the Russian language and provides for the possibility of typical errors when typing text from the keyboard [10,31].

To achieve the stated objective this work addresses the following tasks:

- To formalize the task and to develop a method of associative information retrieval in medical databases, based on determining the degree of similarity of text strings by calculating the Levenshtein distance;
- To develop criteria for the similarity between the words entered and being in the database taking into account the peculiarities of the Russian language and providing for the possibility of typical errors when typing text from the keyboard;
- To implement a method for designing an intelligent interface in medical decision support systems for prescribing drugs.

3 Materials and Methods

3.1 Method and Basic Assumptions

To achieve the objectives in the work are used: methods for determining the degree of similarity of text strings by calculating the Levenshtein distance for organizing an associative retrieval; expert assessment method for developing similarity criteria; methods of phonetic and morphological analysis to take into account the peculiarities of the Russian language; object-oriented approach in the development of software medical decision support system.

3.2 Development of an Associative Information Retrieval Method Based on Determining the Degree of Similarity of Two Strings

Let us consider the features of the implementation of the method for determining the degree of similarity of two strings based on the calculation of the Levenshtein distance [24] in order to implement associative information retrieval in the developing decision support system.

Let two lists be given (two text strings):

$$X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\} \text{ and } X_j = \{x_{j1}, x_{j2}, \dots, x_{jm}\} \quad (1)$$

In general, n and m may be not equal to each other.

From a consideration of two sequences, it is clear that one of them can be transformed into the other. For example, in order to move from string $X_1 = ABC$ to string $X_2 = ABDE$, the first and second characters should be saved, the third one should be replaced and the fourth one should be added. Entering the empty symbol λ , these three variants of transformations can be written in the form of three operations:

$$\begin{aligned} \text{SUBstitution } x_i \rightarrow x_j & \text{ SUB}(x_i, x_j); \\ \text{DESTRUCTION } x \rightarrow \lambda & \text{ DES}(x, \lambda); \\ \text{CREATION } \lambda \rightarrow x & \text{ CRE}(\lambda, x). \end{aligned} \tag{2}$$

Each transformation has its own “price” $c(\dots)$, namely:

$$\begin{aligned} c(x_i, x_j) & \text{ for SUB}; \\ c(x, \lambda) & \text{ for DES}; \\ c(\lambda, x) & \text{ for CRE}. \end{aligned}$$

To estimate the distance between two lists, the concept of the full price of a sequence of transformations is introduced as the lowest of all possible prices that should be “paid” for the transition from the original list to the final one. The distance $\delta(X_i, X_j)$ corresponding to the full price is minimal. This means that if $\delta(X_1, X_2) = c(X_1, X_2)$ then the transition from X_1 to X_2 occurs without intermediate sequences.

The transformation scheme is called the path, usually depicted by arrows. So the transformation path from X_i to X_j is presented in Fig. 1.

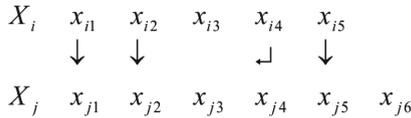


Fig. 1. Transformation from X_i to X_j .

It can be written as a series of operations:

$$\begin{aligned} \text{SUB}(x_{i1}, x_{j1}); \text{SUB}(x_{i2}, x_{j2}); \text{DES}(x_{i3}, \lambda); \text{SUB}(x_{i4}, x_{j4}); \text{CRE}(\lambda, x_{j4}); \\ \text{SUB}(x_{i5}, x_{j5}); \text{CRE}(\lambda, x_{j6}); \end{aligned}$$

Each operation is attributed to a certain price, and the distance between the lists corresponds to the minimum possible price. Let us consider two more lists:

$$\begin{aligned} X_i &= \{x_{i1}, x_{i2}, \dots, x_{in}\}; \\ X_j &= \{x_{j1}, x_{j2}, \dots, x_{jm}\} \end{aligned}$$

Let be $X_i(l) = x_{i1}, x_{i2}, \dots, x_{il}$ and $X_j(k) = x_{j1}, x_{j2}, \dots, x_{jk}$, then, for example,

$$X_i(5) = \{x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}\};$$

$$X_j(3) = \{x_{j1}, x_{j2}, x_{j3}\}.$$

Let be $D(l, k) = \delta [X_i(l), X_j(k)]$. There are three possibilities for moving from $D(l - 1, k - 1)$ to $D(l, k)$:

- by substitution $X_i(l)$ and $X_j(K)$ the price of which corresponds to the price of substitution $SUB(x_{il}, x_{jk})$ of the last elements in each list;
- by creating the last element in the list j , for which the price $CRE(\lambda, x_{jk})$ will have to be paid;
- finally, if a pair $X_i(l - 1)$ and $X_j(k)$ is considered then the price will be , which corresponds to the cost of destroying the last item in the list i .

Briefly, this procedure can be written as

$$D(l, k) = \min \begin{cases} D(l - 1, k - 1) + c(x_{il}, x_{jk}) \\ D(l, k - 1) + c(\lambda, x_{jk}) \\ D(l - 1, k) + c(x_{il}, \lambda) \end{cases} \quad (3)$$

The minimum price corresponds to the optimal path.

Let us to write this in the form of an algorithm:

1. $D(0, 0) = 0$ (initial value)
2. For $l = 1$ to n do
 $D(l, 0) = D(l - 1, 0) = c(x_{il}, \lambda)$
3. For $k = 1$ to m do
 $D(0, k) = D(0, k - 1) = c(\lambda, x_{jk})$
4. For $l = 1$ to n do
 For $k = 1$ to m do
 $d1 = D(l - 1, k - 1) + c(x_{i,l}, x_{jk})$
 $d2 = D(l - 1, k) + c(x_{il}, \lambda)$
 $d3 = D(l, k - 1) + c(\lambda, x_{jk})$
 $D(l, k) = \text{mind}1, d2, d3$
5. Minimum distance $\delta(X_i, X_j) = D(n, m)$

Let us to consider the implementation of the algorithm by example. Let there be two lists:

$$X_i = \{x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}\};$$

$$X_j = \{x_{j1}, x_{j2}, x_{j3}, \},$$

and prices for each operation are set:

$$c(\lambda, x_{jk} = 0.9) \forall k;$$

$$c(x_{il} = 0.8) \forall k;$$

$$c(x_{il}, x_{jk}) = 0 \forall l, k \text{ if } x_{il} = x_{jk};$$

$$c(x_{il}, x_{jk}) = 1 \forall l, k \text{ if } x_{il} \neq x_{jk}.$$

Each list has specific characters:

$$X_i = \{a a b a c\},$$

$$X_j = \{a b d\}.$$

It's clear that $x_{i1} = x_{j1} = a$, $x_{i2} \neq x_{j2}$, etc. The calculated distances $D(l, k)$ are given in Table 1, while the optimal path, in which the procedure (3) is implemented at each iteration step and which leads to the minimum distance, is highlighted in bold underlined in Table 1.

Table 1. Distance values $D(l, k)$

X_j	X_i					
		<i>a</i>	<i>a</i>	<i>b</i>	<i>a</i>	<i>c</i>
	0	0.8	1.6	2.4	3.2	4.0
<i>a</i>	0.9	0	0.8	1.6	2.4	3.2
<i>b</i>	1.8	0.9	1.0	0.8	1.6	2.4
<i>d</i>	2.7	1.8	1.9	1.7	1.8	2.6

From Table 1 we find that $\delta(X_i, X_j) = 2.6$.

The example shows the efficiency of the method, but it does not fully meet the requirements for development. The considered method, analyzing the need for SUB (substitutions) transformation of two compared characters, takes into account only their coincidence or mismatch, which does not quite adequately reflect the measure of closeness of text strings. The paper proposes a modification of the method by entering into its additional criteria of similarity. The development of these criteria is discussed below.

3.3 Development of Criteria for the Similarity of Text Strings Based on the Selected Method

In order for the chosen method to take into account the peculiarities of the Russian language and provide for the possibility of some typical errors when typing words, written in this language, it must be supplemented with new criteria for the similarity between the words (lists) entered and stored in the computer's memory.

As described in the previous paragraph, to detect the degree of similarity of two sequences of characters, the selected method sets the following prices, which should be "paid" for the transition from the original sequence to the final one by converting SUB: $c(x_{il}, x_{jk}) = 0 \forall l, k$ if $x_{il} = x_{jk}$, i.e. at the full coincidence of 2 compared characters; and $c(x_{il}, x_{jk}) = 1 \forall l, k$ if $x_{il} \neq x_{jk}$, at full mismatch of characters. The price of the conversion DES $c(x_{il}, \lambda) = 0.8 \forall l$, and price CRE $c(\lambda, x_{jk}) = 0.9 \forall k$.

Based on the analysis of typical errors in text strings, we introduce additional criteria for detecting the similarity of 2 characters, which are expert estimates of the measure of character proximity:

1. If the compared characters differ only in case (for example, for the Russian language А-а, Б-б, etc.) then the conversion SUB cost is equal to

$$c(x_{il}, x_{jk}) = 0.1 \forall l, k$$

2. The criterion for the detection of similarity, taking into account the possibility of spelling errors during data entry (for vowels of the Russian language о-а, е-и, etc.). In this case, the price of the conversion SUB is equal to

$$c(x_{il}, x_{jk}) = 0.15 \forall l, k$$

If, in addition, the registers of the compared characters do not match either (О-а, А-о, etc.) then the price of the conversion SUB is equal to

$$c(x_{il}, x_{jk}) = 0.25 \forall l, k$$

3. The criterion for detecting the similarity of hard and soft vowels (а-я, у-ю, э-е, etc.). The price of SUB is equal to

$$c(x_{il}, x_{jk}) = 0.2 \forall l, k$$

If additionally there is also a mismatch of registers (А-я, а-Я) then the price of the conversion SUB is equal to

$$c(x_{il}, x_{jk}) = 0.4 \forall l, k$$

4. The criterion is similar to the previous one, but for consonants - the detection of the similarity of voiced and deaf consonants (з-с, б-п, etc.). Respectively for characters with non-matching registers (З-с, з-С, etc.), the prices of SUB are equal to

$$c(x_{il}, x_{jk}) = 0.3 \forall l, k$$

and

$$c(x_{il}, x_{jk}) = 0.5 \forall l, k$$

The similarity criteria described above take into account the specifics of the Russian language. Next, we introduce similarity criteria which provide for the possibility of errors caused by entering information from the keyboard (that is, errors that are most often allowed by operators or computer users during data entry).

5. The similarity criterion, that detects the similarity of characters located next to the keyboard. In this case, the price of the SUB conversion is equal to

$$c(x_{il}, x_{jk}) = 0.35 \forall l, k$$

If there is an additional register mismatch then the price of the SUB conversion is equal to

$$c(x_{il}, x_{jk}) = 0.6 \forall l, k$$

6. The similarity criterion of characters, taking into account the possibility of their ordinal mismatch (for example, по-оп, да-ад, etc.). Accordingly, the prices are equal to

$$c(x_{il}, x_{(i+1)l}, x_{jk}, x_{(j+1)k}) = 0.7 \forall l, k \text{ and } c(x_{il}, x_{(i+1)l}, x_{jk}, x_{(j+1)k}) = 0.75 \forall l, k$$

in case there is also a register mismatch (По-Оп, etc.).

The method of detecting the similarity of text strings, supplemented by these criteria, is used to solve a development problem.

4 Experiment, Results and Discussion

The method of associative information retrieval is used in the decision support system developed by the authors when prescribing drugs in dermatology [4]. When forming a request for the system, the information is entered by the user in text form from the keyboard, and then errors are possible with such input. A mismatch even in one symbol will lead to the fact that the necessary information from the database will not be found. This problem is especially relevant when entering a diagnosis since the system needs to know exactly for which disease it is necessary to select drugs that provide the necessary pharmacological actions. Therefore, the system conducts an associative retrieval on the field «Diagnosis» which allows finding diseases in the database even if there are some errors in the query word.

The input to the associative retrieval procedure is two text strings, one of them is the query X_i , and the other is information from the database «Diagnoses» – a set of text strings $\{X_j\}_k$ where k is number of database entries. The degree of similarity of the analyzed database rows to a query is determined by successively reviewing these rows $l = \overline{1, k}$ and calculating the conversion price according to (3) taking into account the similarity criteria considered earlier. Since the price of conversion $\delta(X_i, X_j)$ depends on the length m of the string X_j from the database, the normalized confidence value of the associative recognition of the query X_i is entered:

$$P_{ij} = (1 - \delta(X_i, X_j)/m) \cdot 100\%$$

The result of an associative retrieval is the most reliable record of the database with which further work is performed. Moreover, if the confidence of such recognition is less than 50%, then the system issues a message about the absence of the requested information in the database.

The developed system conducts such a search in the database of diagnoses of dermatological diseases which contains information about the following types of patient diagnoses: the main ones (in bold) and related ones (Table 2, the diagnoses are indicated in Russian).

In Table 3 shows the results of recognition of one of the diagnoses available in the database (“Psoriasis” – Russian transliteration) with various types of distortions of the query word that may occur when entering this word from the keyboard.

Table 2. Record list of database «Diagnoses»

№	Name	№	Name	№	Name
1	Psoriasis	8	Chronic pancreatitis	15	Hypertonic disease
2	Drug disease	9	Cholelithiasis	16	Angina pectoris
3	Eczema	10	Chronic hepatitis	17	Ischemia
4	Diffuse toxic itching	11	Chronic colitis	18	Arrhythmia
5	Stomach ulcer	12	Chronic tonsillitis	19	Vegetative dystonia
6	Chronic gastritis	13	Chronic pharyngitis	20	Varicose Symptom Complex
7	Chronic cholecystitis	14	Chronical bronchitis

Table 3. Record list of database «Diagnoses»

Query	System response	Reliability, %	Revealed errors
Psoriaz	Psoriaz	100	Full match
psoriaz	Psoriaz	99	1 register mismatch (big-small character)
PSORIAZ	Psoriaz	67	6 big-small characters
Psareaz	Psoriaz	96	2 spelling errors
Psoraz	Psoriaz	90	1 missing character
Psoreazz	Psoriaz	90	1 extra character
Psoareazz	Psoriaz	78	2 extra characters
Psoriyaz	Psoriaz	97	paired vowels
Psorias	Psoriaz	96	voiced-deaf consonants
Psoriah	Psoriaz	91	characters that are side by side on the keyboard
Psoraiz	Psoriaz	79	ordinal mismatch
sporiaz	Psoriaz	78	1 big-small character + ordinal mismatch
Psarviaz	Psoriaz	86	1 spelling error + 1 extra character
porreaz	Psoriaz	51	1 big-small character + 1 missing character + 1 extra character + 1 spelling error
psagariit	There is no such diagnosis in the database	38	1 big-small character + 1 spelling error + 6 wrong characters

As it can be seen, in the first column of Table 3 generated requests to the text field of the medical database with distortions that correspond to typical human operator errors during entering text information from the keyboard are shown. It should be noted that the induced distortions of the query word correspond to the previously developed criteria for the similarity of text strings (the cost of transformation during calculating the Levenshtein distance). As noted earlier, the developed similarity criteria are expert assessments that can be adjusted both during test checks and during further operation of the system.

During test checks, the correction of expert assessments was carried out as follows. For each criterion, a corresponding distortion of the query word is

introduced. After that, the reliability of recognition which was calculated by the system (the third column of Table 3) is compared by the expert with the assessment of the “similarity” of text strings by a person on an intuitive level.

In the case of significant differences in the marked assessments, the corresponding correction of the criterion is performed in order to reduce the differences. Previously, the criteria were introduced after the specified correction. Thus, the results of the test check of the associative search for information in the text field of the medical database confirmed the efficiency and effectiveness of the developed method.

5 Conclusions and Future Work

Based on a review of standard methods of associative information retrieval in text fields of medical databases, the relevance of developing new associative search methods that use various methods for detecting the similarity of text strings is substantiated for designing medical decision support systems with an intelligent interface.

The task is formalized and a method of associative information retrieval in medical databases is developed, based on determining the degree of similarity of text strings by calculating the Levenshtein distance, which is the minimum cost of converting one text string (query) to another (database text field). It should be noted that the cost of converting SUB (substitution) of two compared characters takes into account only their complete match or mismatch, which does not quite adequately reflect the measure of similarity of text strings. The paper modifies the method by developing additional criteria for the similarity of two symbols (substitution transformation cost).

An expert assessment was made of text field errors caused by the peculiarities of the Russian language (case mismatch, spelling errors during entering hard and soft vowels and voiced and unvoiced consonants) and typical text input errors from the keyboard (erroneous input of characters located nearby on the keyboard, ordinal mismatch in a row running characters).

Based on the performed expert assessments, additional similarity criteria (transformation cost for calculating of the Levenshtein distance) between the entered words and words in the database have been developed, which take into account the peculiarities of the Russian language and provide for the possibility of typical errors during entering text from the keyboard.

The developed method of associative information retrieval in medical databases was implemented when designing an intelligent interface in medical decision support systems for prescribing medications. Associative search is performed during entering data from the keyboard and during forming a request to the “Diagnosis” text field.

A test recognition of one of the diagnoses which are available in the database with various types of query word distortions that may occur during entering this word from the keyboard was carried out. In this case, the correct recognition of the request is performed in the case when the distortion reaches 50%. Test

verification confirmed the efficiency and effectiveness of the developed method. Further research is aimed at adapting the method for using other languages.

References

1. Medical information system “Imed”. <https://imed.co.ua/>
2. Andreev, A.M., Berezkin, D.V., Brick, A.V.: Linguistic processor for information retrieval system (2016). https://inteltec.ru/publish/articles/textan/art_21br.shtml
3. Andreev, A.M., Berezkin, D.V., Simakov, K.V.: Model of knowledge extraction from natural language texts. *Inf. Technol.* **12**, 57–63 (2007)
4. Dmitrienko, V.D., Povoroznyuk, O.A.: Multi-criteria evaluation of drugs. *Inf. Technol. Comput. Eng.* **3**, 144–148 (2009)
5. Dudar, Z.V., Khaprov, V.S., Musinov, A.V.: Meta contextual search in the internet. *Eastern-European J. Enterp. Technol.* 104–107 (2005)
6. Graber, M.: SQL for mere mortals. “LORF”, Moscow (2014)
7. Honkela, T.: Self-Organizing Maps in Natural Language Processing. Ph.D. thesis, Helsinki University of Technology, Espoo, Finland (1997)
8. Kondratiuk, S., Krak, I.: Dactyl alphabet modeling and recognition using cross platform software. In: *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018*. vol. 8478417, pp. 420–423 (2018). <https://doi.org/10.1109/DSMP.2018.8478417>
9. Kondratiuk, S., Kruchynin, K., Krak, I., Kruchinin, S.: Information technology for security system based on cross platform software. In: Bonča, J., Kruchinin, S. (eds.) *Nanostructured Materials for the Detection of CBRN*. NSPSSACB, pp. 331–339. Springer, Dordrecht (2018). https://doi.org/10.1007/978-94-024-1304-5_25
10. Kozlovska, T.I., Pavlov, S.V.: *Optoelectronic Means of Diagnosing Human Pathologies Associated with Peripheral Blood Circulation*. LAP LAMBERT Academic Publishing, Beau Bassin 71504, Mauritius (2019)
11. Krak, I., Kondratiuk, S.: Cross-platform software for the development of sign communication system: dactyl language modelling. In: *Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017*. vol. 1,8098760, pp. 167–170 (2017). <https://doi.org/10.1109/STC-CSIT.2017.8098760>
12. Krak, I.V., Kryvonos, I.G., Kulias, A.I.: Applied aspects of the synthesis and analysis of voice information. *Cybern. Syst. Anal.* **49**(4), 589–596 (2013). <https://doi.org/10.1007/s10559-013-9545-9>
13. Krak, Y.V., Barmak, A.V., Bagriy, R.A., Stelya, I.O.: Text entry system for alternative speech communications. *J. Autom. Inf. Sci.* **49**(1), 65–75 (2017). <https://doi.org/10.1615/JAutomatInfScien.v49.i1.60>
14. Krak, Y.V.: Method for construction of dynamics equations for manipulating robots in numerical-analytical form. *Izvestiya Akademii Nauk. Teoriya i Sistemy Upravleniya* 1, 137–141 (1993)
15. Krak, Y.V.: Robot-manipulator’s dynamics: numerical-analytical method of formation and investigation of calculation complexity. *Problemy Upravleniya I Informatiki (Avtomatika)* **2**, 139–148 (1998)
16. Kryvonos, I.G., Krak, I.V., Barmak, O.V., Bagriy, R.O.: New Tools of Alternative Communication for Persons with Verbal Communication Disorders. *Cybern. Syst. Anal.* **52**(5), 665–673 (2016). <https://doi.org/10.1007/s10559-016-9869-3>

17. Kryvonos, I.G., Krak, I.V., Barmak, O.V., Bagriy, R.O.: Predictive Text Typing System for the Ukrainian Language. *Cybern. Syst. Anal.* **53**(4), 495–502 (2017). <https://doi.org/10.1007/s10559-017-9951-5>
18. Kuznetsov, M.A., Nguyen, T.: Mathematical models of information retrieval of web-resources, *Caspian magazine: management and high technology. Syst. Anal. Math. Model.* **22**, 25–30 (2013)
19. Kvyetnyy, R., Bunyak, Y., Sofina, O., et al.: Blur recognition using second fundamental form of image surface. In: *Proceedings of the SPIE 9816, Optical Fibers and Their Applications*, vol. 98161A (2015)
20. Kvyetnyy, R., Sofina, O., Olesenko, A., et al.: Method of image texture segmentation using laws' energy measures. In: *Proceedings of the SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 1044561* (2017). <https://doi.org/10.1117/12.2280891>
21. Kvyetnyy, R.N., Sofina, O.Y., Lozun, A.V., et al.: Modification of fractal coding algorithm by a combination of modern technologies and parallel computations. In: *Proceedings of the SPIE 9816, Optical Fibers and Their Applications 2015. vol. 98161R*. <https://doi.org/10.1117/12.2229009>
22. Lande, D.V.: *Search for knowledge on the Internet. Dialectics*, Moscow (2005)
23. Lande, D.V., Snarsky, A.A., Bezsudnov, I.V.: *Internetika: Navigation in complex networks: models and algorithms*. Book House "LIBROKOM", Moscow (2009)
24. Levenshtein, V.I.: Binary codes with corrections for fallouts, insertions and substitutions of characters. *Rep. Acad. Sci. USSR* **163**(4), 846–848 (1965)
25. Mahmood, A., Safa, A.: Document retrieval systems. *J. Eng. Res. Appl.* **6**(7), 20–22 (2016)
26. Povoroznyuk, A.I.: *Decision support systems in medical diagnostics, Synthesis of structured models and decision rules*. LAP LAMBERT Academic Publishing GmbH & Co. KG, Saarbrücken Germany (2011)
27. Povoroznyuk, A.I., Filatova, A.E., Zakovorotniy, O.Y., Kuzmenko, Y.V., Wojcik, W., Kalimoldayeva, S.: The functional model of instrumental examination of a patient. *Information Technology in Medical Diagnostics II*, London, UK, pp. 253–266. CRC Press (2019). <https://doi.org/10.1201/9780429057618-30>
28. Shuklin, D.E.: Development of a natural language text processing system based on semantic neural network. In: *Materials of the International Scientific-Technical Conference "Artificial Intelligence - 2002"*, vol. 2, pp. 310–314 (2002)
29. Tarasenko, V.P., Mihailyuk, A.Y., Zabolotnaya, T.N.: Context-associative approach to automated spelling correction. *Syst. Res. Inf. Technol.* **3**, 21–30 (2009)
30. Walenda, N.A., Samofalov, L.D., Pavlenko, N.A.: Analysis of phrases of the Russian language based on LR-grammar. *Inf. Process. Syst.* **8**(145), 69–72 (2016)
31. Wójcik, W., Pavlov, S., Kalimoldayev, M.: *Information Technology in Medical Diagnostics II*. Taylor & Francis Group CRC Press, Balkema book, London (2019)
32. Wójcik, W., Smolarz, A.: *Information Technology in Medical Diagnostics*, July 11. CRC Press (2017). <https://doi.org/10.1201/9781315098050>



Analysis of Deep Learning Methods in Adaptation to the Small Data Problem Solving

Iurii Krak^{1,2}(✉) , Vladyslav Kuznetsov¹ , Serhii Kondratiuk^{1,2} ,
Larisa Azarova³ , Olexander Barmak⁴ , and Pavlo Padiuk⁴ 

¹ Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
yuri.krak@gmail.com

² Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv, Ukraine

³ Vinnytsia National Technical University, Vinnytsia, Ukraine
le-azarova@vntu.edu.ua

⁴ Khmelnytskyi National University, Khmelnytskyi, Ukraine

Abstract. This paper discusses a specific problem in the study of deep neural networks - learning on small data. Such issue happens in situation of transfer learning or applying known solutions on new tasks that involves usage of particular small portions of data. Based on previous research, some specific solutions can be applied to various tasks related to machine learning, computer vision, natural language processing, medical data study and many others. These solutions include various methods of general purpose machine and deep learning, being successfully used for these tasks. In order to do so, the paper carefully studies the problems arise in the preparation of data. For benchmark purposes, we also compared “in wild” the methods of machine learning and identified some issues in their practical application, in particular usage of specific hardware. The paper touches some other aspects of machine learning by comparing the similarities and differences of singular value decomposition and deep constrained auto-encoders. In order to test our hypotheses, we carefully studied various deep and machine learning methods on small data. As a result of the study, our paper proposes a set of solutions, which include the selection of appropriate algorithms, data preparation methods, hardware optimized for machine learning, discussion of their practical effectiveness and further improvement of approaches and methods described in the paper. Also, some problems were discussed, which have to be addressed in the following papers.

Keywords: Small data · Deep neural networks · Machine learning · Data dimensionality reduction · Anomaly detection · Data augmentation · Algorithm stability · Tensorflow · Directml

1 Introduction

One of the main directions of improving the work of modern algorithms of Artificial Intelligence (AI) based on the ideology of Big Data [3] is the use of large datasets, large data centers (computing clouds) and, accordingly, increasing the depth and width of the layers of neural networks and other constructs used in deep learning. This, in turn, creates new advances, but they become inaccessible to the individual researchers and AI developers: first, even a pre-trained model with billions of parameters (such as GPT-3) requires a device to scale this model to new data, and, secondly, the repetition (reproduction) of experiments by other studies becomes possible only for those who have a small supercomputer at the level of a small company in the field of artificial intelligence. If this is done by an individual researcher, the threshold for entering this field becomes insurmountable and the next question arises - do one really need big data to summarize its properties and teach an algorithm with such many parameters? There is an unambiguous answer to this question, but it is partly contained in the history of other methods of intelligent data processing, such as the singular value decomposition and the method of group argumentation. In some cases, close to ideal, several dozen data samples were enough to teach such methods. This example is a possible answer to the question: how to make Big Data and Deep Learning is available to a wide range of researchers? If there is an example of successful learning of some algorithms on small data samples, it means that it is possible to scale the model for deep learning methods so that they can be applied to small data samples. This gives a potential leap for research in this field and makes such methods and models available to a wider range of researchers. In order to address this issue the paper covers both theoretical and practical aspect in software and hardware implementation of these algorithms for particular tasks of learning on small data.

Paper Structure. The main results are presented in paragraphs 2–5, as follows:

- The *paragraph 2* discusses related works on machine learning and deep learning, including our works on computer vision, data classification, natural language processing and others. Based upon the problems discussed in that paragraph, the goals and research tasks of the study are presented.
- The following *paragraph 3* discusses pipeline of data preparation and 4 main techniques in data preparation such as anomaly detection, data augmentation, detection of perturbations in data and their affect on the efficiency of learning.
- *Paragraph 4* focuses mostly on the implementation of the algorithms for different processor architectures and underscores the relationship of code optimization for specific architecture and performance of specific algorithm for particular task.
- According to recommendations given in previous paragraphs, *paragraph 5* focuses on the main goals of the study. Based upon these two group of experiments, the graphic data is provided to illustrate different aspects of supervised and unsupervised deep learning, in particular learning on data, denoising and feature representation.

2 Related Works and Problem Statement

A typical machine learning scheme in approaches that use general-purpose machine learning and in-depth learning tend to differ in substance. If we very briefly define the essence of the first - a cascade of algorithms, the input of which receives data (raw data or processed features), and the output - hypotheses about the belonging of a data element to a particular class. In contrast, the concept of deep learning involves the construction of a neural network architecture that combines different basic elements - ordinary (shallow) layers, convolution layers, thinning, recurrent layers and more. Accordingly, the deep network, with some exceptions, can be constructed simultaneously as one algorithm, even if the structural layers of the network are responsible for different functions - detection of features, grouping, and classification.

Let's focus in more detail on two typical learning pipelines that we used in previous studies [17–19] and which have proven themselves on the example of machine learning (classification of social and textual information) and deep learning, respectively (on the example of classification of images of emotional expressions on the face). A typical machine learning model consists of the following methods:

- transformations of data - e.g. singular value decomposition and integral transformations [1, 7, 31];
- grouping of features - e.g. T-stochastic neighbor embedding [5, 30];
- space compaction - e.g. variation and denoising auto-encoders [10];
- classification in the space of reduced dimension - e.g. decision trees, support vector machines and other methods [11, 21].

As one can see from the structure of this model, it is possible to perform training step by step to achieve sufficient accuracy in the reconstruction of feature-space, the number of errors of the 1st and 2nd kind, and so on. The most time-consuming method is the method T-stochastic neighbor embedding, as it builds a nonlinear hypothesis about the relative position in the feature-space of reduced dimension; the second execution time is the method of compaction of the space of features, which is essentially a method of deep learning, but due to fewer layers in the context of the task (compared to the task purely deep learning), it is similar to a normal feed-forward neural network. In contrast to the methods of grouping features and compacting the space of features, methods of dimensional reduction are very fast and can be faster than similar methods of deep learning. The most successful results in terms of execution time and efficiency before the emergence of big data were achieved by combining these two groups, where the classification methods consisted of support vector machines and decision trees [12].

If we analyze such a model, we can say that it is possible to build a certain pipeline of deep learning based on neural network constructs and general purpose machine learning, including the same functions - respectively, the dimensional reduction and classification. The difference, however, is that these methods show greater efficiency with increasing data volumes and, accordingly, an increase in

the number of parameters taken into account in the model [15]. In the context of this study, it is important to show threshold (borderline) examples where the use of deep learning methods is appropriate (small data) and, conversely, where it is more appropriate to use a conventional model of machine learning with a cascading combination of classifiers and dimensional transformation methods.

The main goals of the study are: *1) to identify where are the issues in learning on small data; 2) to investigate practical performance of deep learning algorithms for specific tasks - learning on features and learning on raw data.*

Thus, in order to approach these goals, the datasets should represent different sizes in order to identify the borderline performance for a specific dataset in order to get the best decision plane for classification. This approach, in contrary to studying big datasets, needs less resources, which, as an effect, tends to decrease the computation costs in order to process the data in such way.

Taking in account the problems in small data analysis, **the research tasks are:**

- to create a few datasets which represent different research topics - data preparation, data grouping, clustering and classification;
- to study hidden feature representation in different methods of dimensional reduction, grouping of features and clustering;
- to analyze locations of the biggest density in context of data preparation;
- to study the latent feature space representation and compare such representation with data dimensional reduction methods;
- to conduct tests on algorithms for data classification and clustering.

3 Data Preparation in Machine Learning and Deep Learning

3.1 General View on the Pipeline of Data Preparation

The process of automated data preparation in the two approaches is similar, but in the case of the classic machine learning has certain steps in the analysis of small samples. In this case, the visual analysis of data comes to the fore, which allows assessing the representation of data elements and perform data engineering. Foremost, these are procedures for the selection of informative features, which in combination allows to effectively allocating data classes and conduct classification and clustering procedures [4, 20]. Secondly, it is the analysis of data clusters, where it is possible to identify the main classes and sub-classes of data. In contrast, when learning from data in the “deep” approach, layers contain hidden information that cannot always be used to make decisions about both a feature set (the number of features in hidden network layers) and data sets.

In fact, decisions are ultimately made on qualitative indicators - on the convergence of the algorithm and the large number of runs of training procedures, in order to determine the optimal architecture. Except for layer-by-layer learning which to involve deep auto-encoders to initialize network weights (unsupervised

learning) [2, 28], the informativeness of the features and the data elements themselves is determined by successful data selection (balancing classes in datasets), using numerous samples formed by affine transformations and data augmentation. Let's focus on the most important approaches in data preparation and its representation for visual analysis.

3.2 Detection of Anomalies in Data

Data anomalies usually occur for several reasons: there is an imbalance of classes, the elements of dataset are scattered or contain poor quality data samples, the presence of hidden relationships between features in the data, non-linearity of feature-space representation that cannot be correctly compacted into reduced dimension and others. This, in turn, means that data elements in the feature space are located at a great distance from the centers of classes outside the areas of data crowding and interfere with the area of crowding belonging to other data classes.

This scenario can be given by an example. The properties of datasets and its features do represent hidden properties only if using specific approaches; hence, the points within original dimension can be very scattered and have a lot of dimensions within feature space. This can be outcome using data dimensional reduction, though they may be highly affected by the size of the dataset; for instance, if the dataset is small the points are scattered, but if it is quite big they form clusters (Fig. 1).

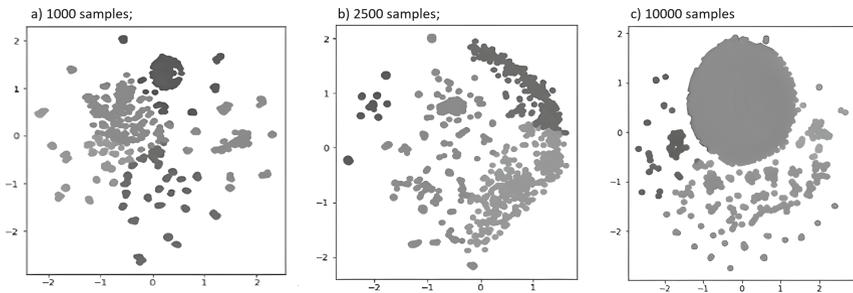


Fig. 1. Evolution of T-SNE feature-space representation using a dataset of reduced number of samples and a full dataset of 10000 samples

According to our investigation, using such quite simple example (shown on Fig. 1), using the method of T-stochastic neighbor embedding on the datasets of different sizes, but containing same elements (e.g. 10%, 25% and 100% as of Fig. 1a, 1b, 1c respectively), one can see, that data points begin to shift one towards another, what, in case of augmented dataset may tend to forming another clusters. This interesting feature was found during our studies dedicated to the processing of scientific texts.

This particular example can be explained by following statement. Since this method, in essence, builds such dependencies that in the new, transformed feature-space, with a relatively large separation band, there is a non-linearity of representations, that can be achieved in relatively large sizes of the datasets. So, having a clusters with relatively large separating band, it is equally possible to use as conventional methods of machine learning and the so-called “wide” neural networks, so as the “deep” and “regular” approach may met. However, the disadvantage of such a representation is the processing time, which potentially requires machine learning accelerators to build a nonlinear data representation.

Despite the disadvantages, this approach is very useful in visual analysis of the data since all its features are represented in the space of the low dimension and, most important - it helps to highlight the anomalies in the final representation using a singular value decomposition or T-stochastic neighbor embedding. Knowing their location, it is possible to apply filtering of data by the root-mean-square error (if there is a relatively large number of data representatives), and in the second - to determine the main data axes by constructing regression [32]. Thus, it is possible to note areas with higher data density and, accordingly, areas with fewer anomalies (Fig. 2).

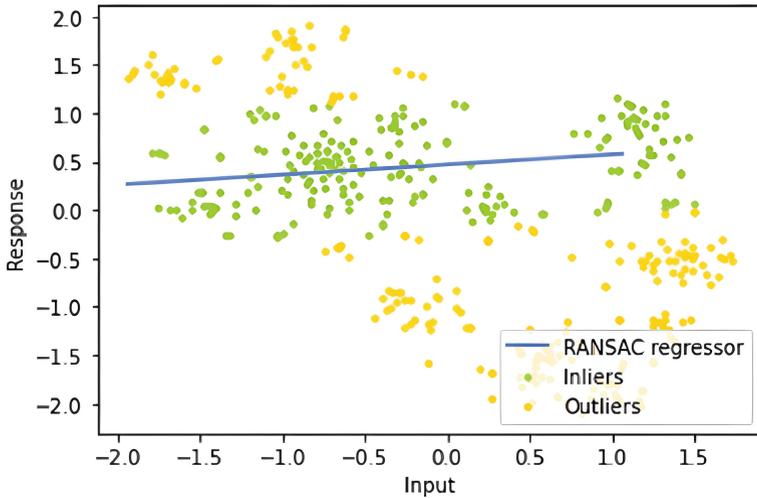


Fig. 2. Principal axis of data and outlier points in reduced feature space

According to the figure (Fig. 2) the feature space representation given by T-stochastic neighbor embedding allows to arrange the data in the same way as usage of principal components. The first principal axis of data shows not only the data crowding, but the property of the dataset itself - e.g. if the data points located very far from the regression also have very big mean-square error they also may represent elements of other classes. Usage of such principle in detail

(for instance, only on one class) may help to find outlier points and identify anomalies in data (if any).

3.3 Data Augmentation

An important step in data preparation is the creation of a data set in which the representatives of the data classes are balanced both in the number of data elements and in their relative location in the case when the data clusters are unevenly distributed (Table 1). To do this, it is possible to artificially generate data samples that have the same location in the feature space (original or reduced dimension), so that the shape of the clusters and the distribution of data samples will remain the same. This can be achieved by applying certain techniques to balanced learning, including SMOTE, ADASYN and others that generate random data samples [6, 8, 9, 22]. Samples with the involvement of specialized unsupervised methods, in particular variation auto-encoders. This subspecies of neural networks contains a core in the middle, which describes the properties of the data, namely their distribution and relative position in the latent space of features, which is generated by the auto-encoder on the output layer of the encoder. Thus, it is possible to generate as a set of random samples in the latent feature space or, conversely, to obtain encoding in the transformed space generated by the source layer of the decoder and thus obtain new data samples (Table 1).

Table 1. Classification rate on test dataset

Precision	Recall	f1-score	Support	Total
Class 1	0.59	1.00	0.74	366
Class 2	0.98	0.95	0.96	454
Class 3	0.90	0.64	0.75	332
Class 4	1.00	0.59	0.74	333
Accuracy			0.81	1485
Macro avg	0.87	0.79	0.80	1485
Weighted avg	0.87	0.81	0.81	1485

In a study on the analysis of scientific texts and the impact of sample size on the quality of the algorithm, including convergence, building an effective hypothesis of data separation by other algorithms (decision trees, support vector machines, etc.), we found that the use of augmentation methods to generate new samples data, and the balancing of data samples in general has a positive effect on the quality of data sampling while maintaining the feature-space configuration of data clusters, their distribution in space and the external boundaries on which the data separation band can be built (Table 2).

Table 2. Classification rate on augmented dataset

Precision	Recall	f1-score	Support	Total
Class 1	0.81	0.97	0.88	1313
Class 2	0.92	0.96	0.94	1239
Class 3	0.95	0.76	0.85	1274
Class 4	0.98	0.93	0.96	1254
Accuracy			0.91	5080
Macro avg	0.92	0.91	0.91	5080
Weighted avg	0.91	0.91	0.91	5080

The comparison of tables listed above (Table 1 and Table 2) may show the difference between number of samples within every class and also their proportion one against another. Change in percentage of each data class as well as increasing the size of the dataset increases the recall in worst-case scenario from 59% on class 1 to 81% and from 87% to 92% which is an impressive result.

Main outcome of such technique is to figure out the minimal size of the dataset, needed to solve a specific problem - studying on small data, not re-creating the dataset from scratch. This may help to see the possible situations where are the deep learning methods can be applied. Also, using same strategy, the other statement can be inferred: if the dataset is separable with desired accuracy in certain feature-space of reduced dimension, it is possible to design a deep learning network that instead of learning from features (representation of lower dimension) can study from the raw data, which, on other hand can be also augmented in original space.

3.4 Perturbation Compensation and Stability of Classification Algorithms

Data preparation always assumes the presence of poor and even corrupted data, which may ultimately affect the construction of a separate plane in hyperspace. As a result of this, the data contains of anomalies, where individual data elements are located outside the main area of data classes, and it affects the hypotheses about data separation towards the location of outlier points, which affects the overall efficiency of classification and clustering algorithms (Fig. 3).

One approach is to generate as much instances of data as possible, so as the “noise” can be studied by the deep learning approach and anomalies can be omitted (which is not the case in the learning on small data). Another and more effective approach is to detect such anomalies and eliminate them by filtering or by compensating for disturbances that increase the displacement of the endpoints of data clusters. This can be achieved by using denoising auto-encoders, which is one of the methods of deep learning. The auto-encoder reinforces the general trends in data - the placement of clusters with high density and, thus, weakens weak trends, which are anomalies and caused by certain disturbances in the data.

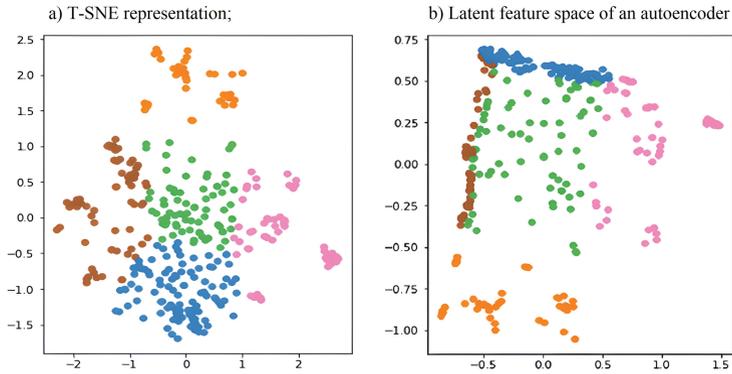


Fig. 3. Comparison of data representations using two different algorithms for data grouping and data dimensionality reduction and their affect on separation band

According to the Fig. 3, it can be seen that the data representation in latent feature space is more “tight” and the separation band becomes very narrow. This is caused by denoising feature of an auto-encoder, because the latent feature space decreases a number of information stored in each feature and thus optimizes for mean-square error of decoded representation and decrease in entropy of the encoded data.

This may give an idea of another application of auto-encoders: since they can reduce the entropy in the data and eliminate disturbances, they may be used to test the algorithm stability of classification algorithms, because in the latent space of features between-class margin and separation band is significantly reduced, which affects the increase of errors of the, except for gradient boosting methods and decision trees, what can be clearly seen on the Fig. 4.

3.5 Orthogonal Transformations in Machine and Deep Learning

The peculiarity of most methods of deep learning and classical neural networks is that the set of features in the hidden layers and, accordingly, the characteristic functions they form are almost equivalent (except for the latest networks such as graph neural network and networks with active involvement of dropout). This means that modifying the architecture to reduce the number of features in the hidden layers is possible only after re-learning the network or (very limited) by visualizing and analyzing images (patterns) of characteristic functions stored in the intermediate layers of the network. In contrast, classical methods, such as Fourier transform, singular value decomposition, wavelets, and others, involve the representation of the original data by a set of linearly independent characteristic functions and allow discarding non-informative features in terms of their power in the final representation. This property is useful when there are hidden links between individual features that affect the redundant information in the features, which ultimately affects the efficiency and speed of classification algorithms.

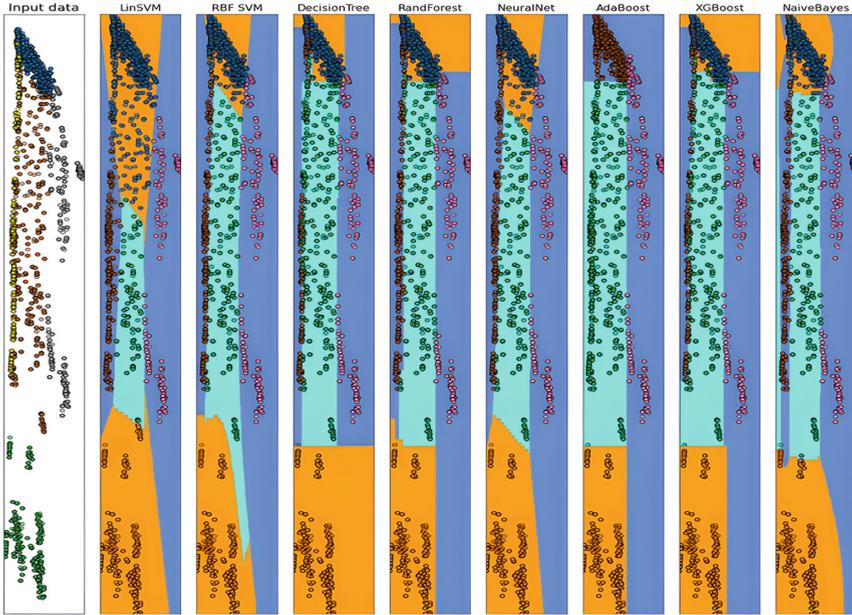


Fig. 4. Hypotheses visualizing for different classifiers in latent feature space

This feature of integral transformations, as a rule, is not applied at construction of architecture of deep networks for unsupervised and supervised learning. Therefore, the dependencies embedded in the network layers are nonlinear and allow hidden linear relationships between features. On the one hand, this allows to build complex nonlinear hypotheses when classifying data (for example, in autoencoders), on the other hand, the presence of restrictions on the weights of the network allows engineering features in the middle of the network and rank them by importance. To over-come these problems, we proposed a modification of the autoencoder, taking into account the limitations on the orthogonality and linear independence of the weights; thus, the representation of data (forms of data clusters) in the latent space of features is very similar to the representation of features obtained by means of a singular value decomposition or similar integral transformations.

4 Note on Implementation of Algorithms for the Task

4.1 Overview of Libraries for Machine Learning Used in Study

Currently, researchers have quite a lot of machine learning algorithms and their implementations for modern personal computers. The most famous libraries are Scikit-learn and TensorFlow [26] and, in particular its newest implementation TensorFlow-DirectML [27] from Microsoft, which enables usage of hardware machine learning acceleration for various hardware vendors (Intel, AMD, Nvidia).

The-se libraries are widely used among researchers and scientists in the field of machine learning and data science due to the widespread use of Python [23]; though, we have to admit that there are many other languages such as C++, Java, C# (Caffe, Theano, Torch, dl4j, Smile and others) used in AI research but in very specific tasks where the software performance is more important. These libraries for various languages can be used either on central processing units (CPU) or graphic processing units (GPU) or both, which mostly depends on their adaptation (OpenCL, CUDA). Because our research was more focused on Python, we will look at our machine learning and deep learning experience for this programming language, in particular its Intel implementation, Anaconda, which includes pre-built Intel MKL libraries for matrix operations.

4.2 Implementations for Processors

Machine learning and deep learning libraries for Python are mostly high-level (C and C++) compiled code that is built as a library (DLL) in the structure of a Python interpreter that executes instructions on a Python listing. In the case of libraries such as Open CV or DLib, the source text is compiled on the user's machine for the processor architecture where the program will be executed and according to a set of processor instructions. In the case of more universal libraries, such as Scikit-learn, a ready-made binary file is loaded, which is optimized for the architecture of the whole processor family, and in the case of code generation can be more optimized for a specific architecture (AMD or Intel). According to a recent study, we tried to evaluate the relationships between the number of CPU cores, its theoretical performance, and actual performance on various machine and deep learning tasks - dimension reduction, feature grouping, classification, and clustering (Table 3) in order to find the most optimal architecture for specific task (data preparation, feature space reduction, classification etc.).

Table 3. Performance indicators of machine learning procedures for different architectures of processors

Procedure name	Max, s	Min,s	Max/Min	Average
Read	1.45	0.139	10.37	0.6081
Reshape	0.006	0.001	6.01	0.0029
Dimensionality reduction	2.173	0.075	28.93	0.6348
Grouping of features	65.57	10.81	6.06	39.07
Clustering of features	0.6	0.072	8.24	0.2387
Sample preparation	0.007	0.002	2.68	0.0051
Deep ANN init	0.269	0.05	4.83	0.1585
Deep ANN learning	117.643	24.10	4.88	71.0883
Deep ANN inference	2.221	0.583	3.81	1.2514
Decision trees training	0.826	0.148	5.59	0.3859

There are a few practical outcomes of the experimental results shown in Table 3:

- the tasks that involve simple manipulations with memory (e.g. reshape of an array) display some difference in performance, but it is insignificant both in absolute and relative means and mostly affected by number of computing cores;
- interestingly, the relative difference (28x) in such task as singular value decomposition is more prominent, than of data grouping or learning of the deep network. It means that the matrix operations including multiplication are performed better on newer versions of processors, though it depends on implementation;
- the practical efficiency of learning rely on the speed of the all components included in the task starting from the reading from disk till machine learning tasks and inference on new data, in case if there is a weak link. For instance, slow operation of grouping of features or deep learning, which are most time consuming tasks may affect the overall time and the proper architecture has to be used in the experiment.

5 Evaluation of the Practical Effectiveness of Deep Learning Methods on Real Data

5.1 Data Used in Experiments and Previous Study in Area

In order to evaluate the algorithms of deep learning, a series of experiments were conducted on different data - textual information, video images, medical data, sound samples etc. (Fig. 5). Some data, in particular scientific texts and video images, were obtained in our research and used to test the concept of deep learning and improve classification methods involving visual analysis of data clusters, transforming the dimensional of data, features creation, face recognition etc. [16, 19]. Let's dwell in more detail on another aspect - the analysis of applicability on different types of data and, most importantly - on the impact of sample size on the overall effectiveness of deep learning methods. The peculiarity of deep neural networks and derived constructs is that they can in some cases work with raw data without the use of third-party methods; they have shown themselves best in the analysis of digital images. Instead, when studying more complex data, such as sound or textual information, the input of the neural network receives already converted information. In our research, the vast majority of information is the feature vectors obtained by analysis of methods of feature extraction (detection) - in the case of analysis of facial expressions we used preprocessing by computer vision, in the analysis of sound information - integral transformations, in the analysis of scientific texts - methods of text mining (Fig. 6).

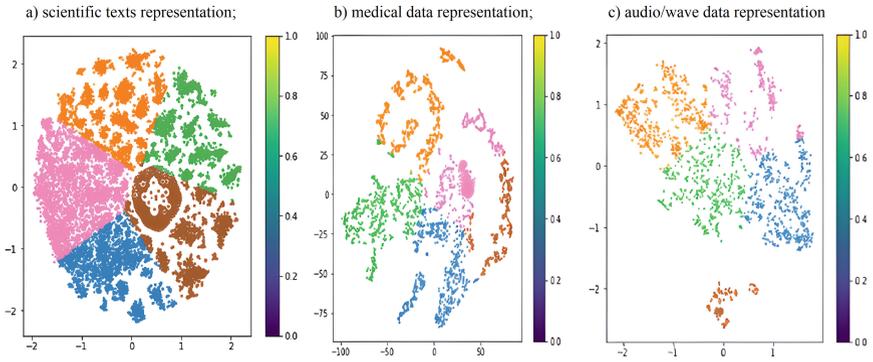


Fig. 5. Data representation in feature space of lower dimensionality for a datasets used in the experimental study

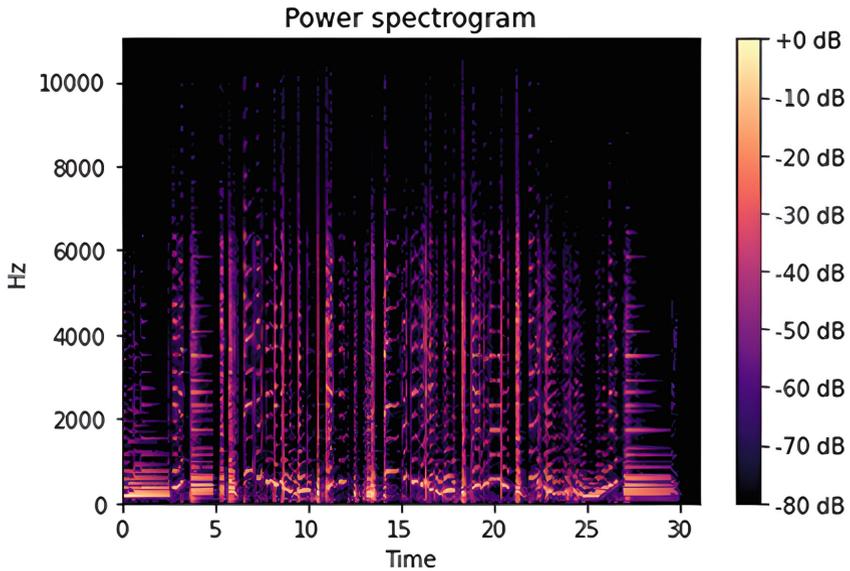


Fig. 6. Fourier transform of sound sample from dataset used in the experiments

5.2 Application of Computer Vision Techniques for Neural Networks

In the work [19], devoted to the study of emotional expressions on the human face, the technology of obtaining samples of trajectories of features on the face with the involvement of computer vision methods was proposed. This allowed to transform the multidimensional space of features (which is the image) on the face into a set of coordinates of the centers of features and to analyze both their static positions (instantaneous states) and changes in their position over time.

We studied such a set of features by methods of dimensional reduction such as singular decomposition and classification methods such as support vector machines. The sample size, which at the time of publication was about 170 samples did not allow the full use of shallow neural networks and even more so - deep learning methods due to slightly worse (in the context of the problem and based on the size of the training sample) performance indicators - errors 1 and 2nd kind; constructs such as the deep belief network [13], the convolution neural network [14,24,25,29], and the varieties of cascading denoising auto-encoders were tested (Table 4).

Table 4. Performance indicators of machine learning algorithms on facial expressions

Type of method	Accuracy and size of vector
Multi-layer neural network	75 %, 400×1
Denoising auto-encoder	98 %, 400×1
Deep belief network	75 %, 400×1
Convolution neural network	75 %, 400×1
Singular value decomposition and decision trees	75 %, 140 eigen values

These experiments (Table 4) showed that on a given number of samples (small dataset), the accuracy of recognition reached up to 95% for the support vector machines method and about 75% for the deep learning methods noticed in above. However, when using the initiation of the weights with a cascading denoising auto-encoder and layer-by-layer learning of shallow neural networks, the accuracy of recognition approached the accuracy of the support vector machines method and indicated the potential for use of auto-encoders to reduce data dimensionality and grouping of features.

5.3 Latent Representation of Features as an Alternative to Data Dimensionality Reduction Methods

The widespread use of integrated methods of data analysis, such as singular decomposition, wavelets, Fourier transform, allowed to present data as a combination of linearly independent characteristic functions, where the first main decomposition coefficients could be used as points on a two-dimensional diagram for visual analysis of the data and, accordingly, the separation of entries into separate classes.

With the advent of constructs such as auto-encoders and their further development in terms of data representation (e.g., linear auto-encoder with constrained weights and variation auto-encoder) it became possible to similarly encode information and obtain a data representation that meets the specified conditions (dimensionality of the feature vector, orthogonality of weights, error of reconstruction) and, accordingly, to have an alternative representation of the

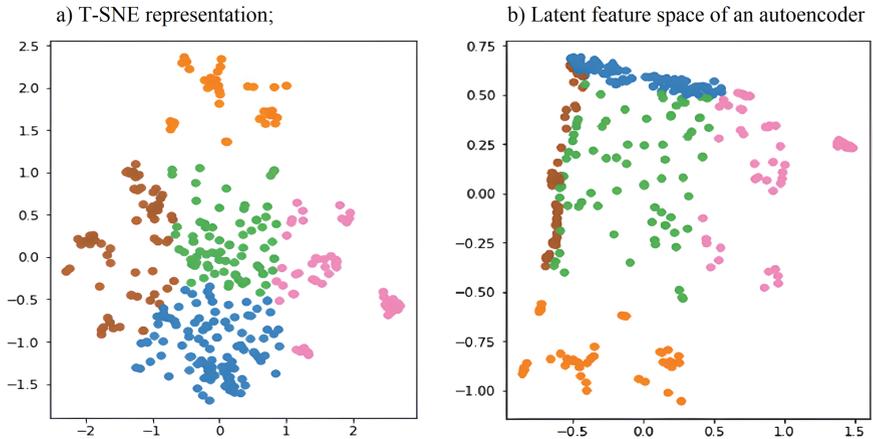


Fig. 7. Comparison of data representations of different methods of dimensionality reduction

data, including nonlinear (Fig. 7). The peculiarity of the autoencoder is that after each iteration a new unique location of data points is built in a space of reduced dimension (which is also called latent feature space), which meets the basic requirements of the optimization algorithm - minimizing mean-square data reconstruction error. In this space, the peculiarities of the auto-encoder are revealed - the distance between the centers of data accumulation decreases and the value of the total and between-the-class standard deviation decreases, which can be used for noise reduction.

As part of the tasks solved in the analysis of scientific texts, facial expressions and other data, we found that the autoencoder can be used for various tasks not directly related to reducing the dimensionality of data - first, visual analysis of data, and secondly (because in the latent space it is possible to compensate for noise), as well as generate new data using variational autoencoders (Fig. 8).

5.4 Performance of Deep Learning Methods on Small and Large Datasets

As part of the experiments to solve the problems of small dataset analysis, both processors and GPU were tested, which differed in both architecture and generation, and which allowed us to assess the performance taking into account the aging of equipment. The main purpose of this experiment was to establish the limit cases of the use of processors and, accordingly, GPUs for the analysis of small data samples.

Based on a series of experiments, we found that data samples known as “toy datasets” do not significantly accelerate learning on the GPUs compared to modern processors, but significantly lose in speed. The execution time of the iteration of the algorithm on CPU is a fixed value. However, with the increasing

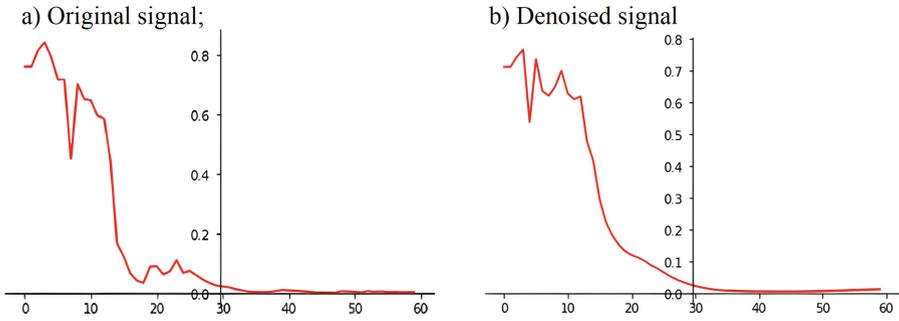


Fig. 8. Denoising effect of an autoencoder on data. The vertical axis represents the relative value of the signal in range [0;1], the bottom axis - the number of measurement (time)

complexity of the calculation there is another trend - a gradual increase in the efficiency of the machine learning accelerator to the estimated maximum value (Table 5).

Table 5. Performance ratio of auto-encoder training in different tasks

Task name	CPU task time, s	GPU task time, s	Performance ratio
Unconstrained AE, small number of iterations	0.46	0.61	0.75
Slightly constrained AE, small number of iterations	2.19	6.97	0.31
Slightly constrained AE, medium number of iterations	27.55	7.36	3.74
Moderate constrained AE, medium number of iterations	32.98	7.68	4.29
Highly constrained AE, high number of iterations	41.11	7.85	5.24

We figured out that the sample size and the degree of fullness of the RAM of the graphics accelerator significantly affects this amount of delay: it consists of the time of reading from disk or RAM and sending (writing) to the memory of the graphics accelerator. In the presence of narrow data bus with the processor, bottleneck effects can occur, where the weak link is no longer the number of computing cores, and memory itself. In the case of exceeding the system memory of the video accelerator and the presence of slow system memory, the video accelerator in almost all cases loses to the processor. Exceptions to this are

cases where these shortcomings are compensated by big number of iterations. In a neural network training experiment dedicated to recognition of individual emotions in a photographic image, we found that execution time affect the overall efficiency ratio to the processor (Table 6).

Table 6. Performance of convolution network training using different hardware

Performance parameter	CPU performance	GPU performance
Number of items	448/448	448/448
Time on epoch 1	449 s	107 s
Time on epoch 2	448 s	22 s
Loss on epoch 1	1.7944	1.7978
Loss on epoch 2	1.4812	1.4888
Accuracy on epoch 1	0.3116	0.3112
Accuracy on epoch 2	0.4364	0.4260

Therefore, as a result of research, we obtained the simple method of definition of the most effective computational means for the task of our research. Initially, the reference AI benchmarking task was calculated, which is slightly smaller in scale, but the amount of system memory involved is satisfactory for running the test task. The next step is to find reference data on the performance of different computing systems on reference tasks, showing the degree to which one device differs from another. Then the target (desired) speed for the given task with a full set of data (operations per second) is selected and the nearest and economically feasible device for this task can be found. In the framework of research we have shown that the theoretical performance rate of machine learning accelerator, calculated from tabular data from open sources and on the results of localization of features on the face is consistent with the actual performance rate, which involves such networks - the classification of images with facial expressions and differs by a small amount for many iterations of the algorithm.

6 Conclusions

The Big Data and Deep Learning is very popular among many researchers in the field of machine learning, but entry in this area is not always possible on a large scale (with really large samples and large models with billions of parameters). That is why acquaintance with it begins with relatively small samples of data obtained by researchers themselves or obtained from the Internet. The rapid development of modern computing devices (processors and graphics accelerators) and the recent rise in their prices due to the global shortage of microelectronics does not always make it economically feasible to use such computing devices, and older generations limit the use of such tools for deep learning.

That is why the tasks of deep learning are scaled to the capabilities of the researcher; this, in turn, poses the problem of analyzing small datasets by deep learning methods. As mentioned in Sect. 5.3, the analysis of small data does not always give satisfactory results by the deep neural networks. Therefore, based on our own experience of developing neurofunctional information transducers, it is more appropriate to carefully study methods that are similar in idea to the methods of reducing the dimensionality of data: since the autoencoders are very similar to singular value decomposition by the idea, it is possible to create dimensional transformations that are not only similar to singular value decomposition, but also have some of its properties (paragraph 3.4).

Subject to the preparation of a sample (for example, involving data augmentation methods), which allows the operation of deep learning methods on a real problem, deep learning can be implemented similarly to classical machine learning methods with step-by-step or cascading operations of extraction of features and classification or clustering by deep neural networks (paragraph 2).

In the following works we plan to investigate more carefully other constructs of neural networks - such as recurrent neural networks and to evaluate the possibility of their application for the problems solved in our research, in particular in the analysis of scientific texts and medical data.

References

1. Albtoush, A., Fernández-Delgado, M., Cernadas, E., Barro, S.: Quick extreme learning machine for large-scale classification. *Neural Comput. Appl.* **34**(8), 5923–5938 (2021). <https://doi.org/10.1007/s00521-021-06727-8>
2. Aloysius, N., Geetha, M.: A review on deep convolutional neural networks. In: 2017 International Conference on Communication and Signal Processing (ICCSP), pp. 588–592. IEEE (2017). <https://doi.org/10.1109/iccsp.2017.8286426>
3. Alzubaidi, L., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **8**(1), 1–74 (2021). <https://doi.org/10.1186/s40537-021-00444-8>
4. Babichev, S., Durnyak, B., Zhydetsky, V., Pikh, I., Senkivskyy, V.: Application of optics density-based clustering algorithm using inductive methods of complex system analysis. In: 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT), vol. 1, pp. 169–172 (2019). <https://doi.org/10.1109/STC-CSIT.2019.8929869>
5. Chan, D., Rao, R., Huang, F., Canny, J.: T-SNE-CUDA: GPU-accelerated T-SNE and its applications to modern data. In: 2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), pp. 330–338. IEEE (2018). <https://doi.org/10.1109/cahpc.2018.8645912>
6. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)* **16**, 321–357 (2002). <https://doi.org/10.1613/jair.953>
7. Dongarra, J., Gates, M., Haidar, A., et al.: The singular value decomposition: anatomy of optimizing an algorithm for extreme scale. *SIAM Rev.* **60**(4), 808–865 (2018). <https://doi.org/10.1137/17m1117732>

8. Hast, A., Vast, E.: Word recognition using embedded prototype subspace classifiers on a new imbalanced dataset. *J. WSCG* **29**(1–2), 39–47 (2021). <https://doi.org/10.24132/jwscg.2021.29.5>
9. He, H., Bai, Y., Garcia, E., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328. IEEE (2008). <https://doi.org/10.1109/ijcnn.2008.4633969>
10. Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., Weinberger, K.Q.: Multi-Scale Dense Networks for Resource Efficient Image Classification (2017). <https://doi.org/10.48550/arXiv.1703.09844>
11. Izonin, I., Tkachenko, R., Gregus, M., Duriagina, Z., Shakhovska, N.: PNN-SVM approach of Ti-based powder's properties evaluation for biomedical implants production. *Comput. Mater. Continua* **71**(3), 5933–5947 (2022). <https://doi.org/10.32604/cmc.2022.022582>
12. Izonin, I., Tkachenko, R., Shakhovska, N., Lotoshynska, N.: The additive input-doubling method based on the SVR with nonlinear Kernels: small data approach. *Symmetry* **13**(4), 1–18 (2021). <https://doi.org/10.3390/sym13040612>
13. Jiang, M., et al.: Text classification based on deep belief network and softmax regression. *Neural Comput. Appl.* **29**(1), 61–70 (2016). <https://doi.org/10.1007/s00521-016-2401-x>
14. Khan, A., Sohail, A., Zahoor, U., Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **53**(8), 5455–5516 (2020). <https://doi.org/10.1007/s10462-020-09825-6>
15. Krak, I., Barmak, O., Manziuk, E.: Using visual analytics to develop human and machine-centric models: a review of approaches and proposed information technology. *Comput. Intell.* 1–26 (2020). <https://doi.org/10.1111/coin.12289>
16. Krak, Y., Barmak, A., Baraban, E.: Usage of NURBS-approximation for construction of spatial model of human face. *J. Autom. Inf. Sci.* **43**(2), 71–81 (2011). <https://doi.org/10.1615/jautomatinfscien.v43.i2.70>
17. Krivonos, Y.G., Krak, Y., Barchukova, Y., Trotsenko, B.: Human hand motion parametrization for Dactilemes modeling. *J. Autom. Inf. Sci.* **43**(12), 1–11 (2011). <https://doi.org/10.1615/JAutomatInfScien.v43.i12.10>
18. Kryvonos, I., Krak, I.: Modeling human hand movements, facial expressions, and articulation to synthesize and visualize gesture information. *Cybern. Syst. Anal.* **47**(4), 501–505 (2011). <https://doi.org/10.1007/s10559-011-9332-4>
19. Kryvonos, I.G., Krak, I.V., Barmak, O.V., Ternov, A.S., Kuznetsov, V.O.: Information technology for the analysis of mimic expressions of human emotional states. *Cybern. Syst. Anal.* **51**(1), 25–33 (2015). <https://doi.org/10.1007/s10559-015-9693-1>
20. Lytvynenko, V., Lurie, I., Krejčí, J., Voronenko, M., Savina, N., Ali Taif, M.: Two step density-based object-inductive clustering algorithm. In: Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education” (MoMLeT and DS-2019), vol. 2386, pp. 1–19. CEUR-WS, Shatsk, Ukraine (2019). <http://ceur-ws.org/Vol-2386/paper10.pdf>
21. Lytvynenko, V., Savina, N., Krejčí, J., Voronenko, M., Yakobchuk, M., Kryvoruchko, O.: Bayesian networks' development based on noisy-MAX nodes for modeling investment processes in transport. In: Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education” (MoMLeT and DS-2019), vol. 2386, pp. 1–10. CEUR-WS, Shatsk, Ukraine (2019). <http://ceur-ws.org/Vol-2386/paper1.pdf>

22. Menardi, G., Torelli, N.: Training and assessing classification rules with imbalanced data. *Data Min. Knowl. Disc.* **28**(1), 92–122 (2012). <https://doi.org/10.1007/s10618-012-0295-5>
23. Python: An open-source programming language, environment and interpreter (2022). <https://www.python.org/about/>
24. Romanuke, V.: An attempt of finding an appropriate number of convolutional layers in CNNs based on benchmarks of heterogeneous datasets. *Electr. Control. Commun. Eng.* **14**(1), 51–57 (2018). <https://doi.org/10.2478/ecce-2018-0006>
25. Sultana, F., Sufian, A., Dutta, P.: Advancements in image classification using convolutional neural network. In: 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 122–129. IEEE (2018). <https://doi.org/10.1109/icrcicn.2018.8718718>
26. TensorFlow: A system for large-scale machine learning (2022). <https://www.tensorflow.org/about/>
27. TensorFlow-DirectML: Github repository for tensorflow fork accelerated by directml (2022). <https://github.com/microsoft/tensorflow-directml>
28. Vahdat, A., Kautz, J.: Nvae: A deep hierarchical variational autoencoder (2020). 1048550/arXiv. 2007.03898
29. Wiatowski, T., Bolcskei, H.: A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Trans. Inf. Theory* **64**(3), 1845–1866 (2018). <https://doi.org/10.1109/tit.2017.2776228>
30. Yona, G., Moran, S., Elidan, G., Globerson, A.: Active Learning with Label Comparisons (2022). <https://doi.org/10.48550/ARXIV.2204.04670>
31. Zebari, R., Abdulazeez, A., Zeebaree, D., et al.: A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *J. Appl. Sci. Technol. Trends* **1**(2), 56–70 (2020). <https://doi.org/10.38094/jastt1224>
32. Zhang, G., Chen, Y.: More informed random sample consensus. arXiv (2020). <https://doi.org/10.48550/ARXIV.2011.09116>



Cognitive and Information Decision Support Technologies for Operational Management of Energy-Active Objects in Boundary Modes

Lubomyr Sikora , Natalya Lysa , Roman Martysyshyn ,
and Yuliya Miyushkovych 

Lviv Polytechnic National University, Lviv, Ukraine
{liubomyr.s.sikora,nataliia.k.lysa,roman.s.martysyshyn,
yuliia.h.miiushkovych}@lpnu.ua

Abstract. The article considers the activity of the operator in emergency situations. It is proved that the implementation of perception of situations (building an image of the situation in terminal time on the basis of intelligent processing of data streams) in the control process in a hierarchical automated control system depends, on the one hand, on the intellectual level of the operator, and on the other hand - on his professional training, knowledgeable training and multimedia presentation tools. The article also considers the operator's ability to perceive scenarios and images of events (based on the processing of data streams), to identify signs of criticality of modes and factors of influence on both the technological system and decision-making (to ensure minimization of risk situations). The most complex component of this process is the perception (as a whole) by the person-operator of a set of automated control system of technological processes structures (projecting 3D image). Such an image should represent physical and chemical-technological energy-active processes in their interconnection, representation of these actions in the cognitive structure of the operator's thinking (in the form of scenario configurations trends), their evaluation when making control decisions. On the basis of the conducted research, the requirements for efficient decision-making were formulated. An analysis of the problem of assessing the operator's intellectual abilities was carried out. The functional structure of the purpose-directed activity process and the scheme of control tasks in conditions of the risk of accidents (which leads to the release of toxic waste into the environment) is given.

Keywords: Intelligence · Neurostructure · Operational · Sensor · Risk · Emergencies · Man-made structures · Decision-making · Management · Cognitive model · Emissions into the ecological environment

1 Introduction

Modern production is a complex integrated man-machine controlled systems. Control strategies (to achieve the goals of functioning) are included both in the structure of the automated control system (ACS), and in the knowledge base and standards of professional skills of the operator. In the hierarchy of ensuring the reliable functioning of the ACS system, the operator has tasks that need to be solved on the cycles of current and terminal time:

- monitoring the dynamic state of power active objects of the automated control system of technological processes (ACS TP);
- assessment of the mode situation of power-generating facilities;
- formation of coordinating actions to maintain target system functioning (both in manual and automatic modes) in case of deviation from the target functioning mode trajectory;
- control and regulation of technological processes in normal modes and emergency situations (according to regulatory requirements and the target task of reducing harmful emissions).

The modern level of information technology and the means of their implementation (computer software, network equipment) allow to implement logically structured algorithms for forming, making and implementing decisions. In the conditions of information and system attacks, this will lead to the failure of the logic of the automated control system and the occurrence of accidents. That is, such a system does not provide (at the cyber level) human-machine interaction of coordination actions of operators (cognitive agents), which due to their intelligence can prevent accidents. Failure to take into account the possibility of a human as a strategic player in the context of threats leads to negative consequences. That is why solving the problem of human (cognitive agent-operator) interaction with an automated control system provides an appropriate level of safety of technogenic system functioning (cybersecurity), and therefore such a problem is relevant.

The operator in the ACS becomes an integral intellectual unit in the structure of the automated control processor, and the reliability of the system depends on it . A characteristic feature of such systems is the distribution of the information load in accordance with the objectives of the ACS TP tasks. This requires processing data flows of different information significance, identifying the characteristic features and functional characteristics of the energy-active control object, modes in the state space and the target space. The operator controls the behavior of the system relative to the target and forms decisions to coordinate the movement of the system in the direction of the target area. This is the basis for building a scenario of events (under the influence of resource and information attacks on the object's state control processes). Data processing processes and decision-making procedures increase the mental tension of the operational staff, which can lead to erroneous decisions with unreasonable risk. That is, according to the situation, the price of error increases. Therefore, it is justified to form a set of requirements for both the level of intellectual readiness and psychophysiological characteristics of the operator and ACS TP (as a process control system).

When implementing the control process in a hierarchical ACS, the perception of the situation, the construction of an image of the situation in terminal time (based on intelligent processing of data flows) depends on one hand on the intellectual level of the operator, and on the other hand - on the professional training and level of knowledge. The operator's ability to perceive scenarios and images of events (based on the processing of data flows), to identify signs of critical modes and action factors of influence (both on the technological system and decision-making) ensures the minimization of risk situations.

2 Problem Statement

The problem of situational stability of man-made structures under conditions of complex threats. The solution to this problem is the development and justification of methods and tools using information technology for intelligent data processing. This would provide a high plausibility of the results as a basis for reducing the risk (in the event of disturbances, information and resource attacks) of making the incorrect target decisions by operational personnel in automated control systems operating in limiting modes (which include energy-active objects in their structure). Accordingly, in the structure of the problem can be divided into several decision-making tasks that are necessary for the formation of strategies to achieve current and strategic goals (under the influence of information and resource disturbances). This allows to minimize the risk based on:

- cognitive support for decision-making (due to a more effective method of assessing the essence of the situation);
- analysis of problem situations (due to information technology of fuzzy data processing);
- effective operator activity (by increasing the ability of the person's intelligence to act);
- risk assessment (due to a clear interpretation of the situations).

The purpose of the research is to develop an effective system to support operational decision-making in an automated control system at risk based on the integration of cognitive methods, information technology and system analysis for the synthesis of strategies to achieve the goal and the stable functioning of the control system.

Object of Research. The processes of imaginative perception of heterogeneous data flows by the cognitive system of the ACS operator, their elaboration, assessment of situations to identify indicators of technical and managerial risks and triggers.

Subject of Research. Models, methods and information technology of assessment of situations, based on heterogeneous data on the object by the cognitive system of the operator. This is necessary to make decisions in the process of solving problem situations (which arise in the system under the action of perturbations and information attacks).

Research Methods. The problem of effective and sustainable management of energy-active objects in the structure of integrated technogenic systems was investigated by the following methods:

- system analysis of the organization of the functioning of the structure of objects and decision-making processes;
- system methods of technogenic risks assessment;
- information technologies of selection and processing of data on the state of objects;
- cognitive psychology of thinking processes in making management decisions under conditions of threats and fuzzy data on the current situation in the system.

The main contribution of the article is to study the impact of the human operator cognitive component on the ACS TP structure in stressful situations and emergencies.

3 Literature Review

The analysis of the complex problem of studying control processes in automated systems with energy-active objects was carried out taking into account the literature sources on the subject.

The authors' experience in the process of project implementation and research work at the Burshtyn Heat and Power Plant (Ivano-Frankivsk region, Ukraine) was also taken into account. These projects were carried out within the framework of the complex problem of ensuring the safety of the power unit and relate to the development of laser systems for measuring the concentration of coal dust (supplied to the boiler combustion zone as the main generator of thermodynamic energy to create high pressure steam for turbines). The problem was that emissions of coal dust concentrations above the limit value could lead to the creation of explosion factors in the combustible mixture in the volume of the heat generation reaction zone. Laser systems designed according to the project and implemented on the power unit, provided continuous control of concentration in accordance with regulations and signaling the approach to critical levels of danger. These systems have become one of the indicators of the crisis situation for operators to assess in the event of an emergency.

Many modern literary sources are devoted to the problems of man-machine interaction. In the book [4] the peculiarities of such systems are considered. The peculiarities of the design of systems for which the most important condition is safety are described.

The influence of psychological factors and peculiarities of human operator perception on the work environment and decision making are given in [15]. The work [14] is devoted to the problems of perception of visual images by a human operator. Psychological and cognitive aspects of man and their impact on the tasks are considered in the works [5, 6].

The work [3] is devoted to the problems of decision-making in energy-active systems. Also, this problem has been repeatedly raised in the works of the authors.

The works were devoted to the peculiarities of perception [10], the peculiarities of cognitive psychology [11], the problems of risk assessment in emergency modes of systems [12]. Works [1, 13] are also devoted to the problem of risks.

In the collective works of the authors the complex of problems of management of integrated systems on the basis of methods of the system analysis and cognitive psychology [6, 10] on the basis of integration with information technologies is considered. The role of cognitive characteristics in decision making is emphasized.

Problems and features of ACS design with potential operator influence are considered in [9]. The works [2, 7, 8] are devoted to the analysis of problems and organization of large technical systems. Much attention to decision-making in a multicriteria is paid in the works [16].

The main aspects of the problem of managing energy-active systems are considered in mentioned works that concerning the components of intellectual and operator activity. The functions and role of cognitive structure in the process of managerial personnel activity in decision making are not fully investigated and the problem is not solved. In this regard, the direction of such research in the future are relevant. The research presented in the article is devoted to this unsolved problem.

4 Materials and Methods

4.1 Analysis of Problematic Tasks of Selection and Cognitive Processing of Data in Making Target Decisions on the Control of the ACS Operator

The structure of an energy-active object and resource flows are affected by the actions of active influences and information attacks on control processes. The operator without special training can not make adequate decisions in stressful situations, because it is necessary to perform a set of operations of intellectual processing of data on the object state.

When the operator makes decisions under extreme conditions, it is necessary to perform the following operations to solve the problem of assessing the situation, developing a goal, decomposition of tasks (having information and cognitive nature in decision-making conditions):

- selection, processing and evaluation of data on the status of objects in the structure of the ACS TP;
- interpretation of data, identifying the content of the data (based on the procedure of classification in the main space of the system according to the multiplicity of signs of the mode of functioning of objects);
- determination of the degree of deviation of the system state from the target state (and based on the specified strategies, develop a procedure for taking management decisions, tactics, and plans of command actions);
- develop tactics for a plan of command operations (in accordance with the management strategies for implementation of the objectives);

- to provide a method for forming dynamic decision-making tables (for planning the sequence of command goal actions);
- to assess management risks under conditions of threats.

On the basis of the analysis justified the choice of management intellectual and information operations of processing data flows about the state of the object. An operating model of performing goal-oriented management decisions and implementing actions (which bring the technogenic system into the target area of the mode of functioning under disturbances and attacks) was developed.

In the complex, this problem of managing complex systems is not solved, as more attention is paid to the algorithmic structure of the decision-making process for the management of the aggregate structure. The cognitive component in stressful situations (as a source of possible errors in the formation of executive teams) is not taken into account, which becomes the first cause of accidents.

In the process of management (at the system, resource, information and operational levels) the tasks of goal-oriented behavior of staff at the psychophysical and cognitive level of situation analysis and decision-making are solved. In extreme conditions, this, in turn, is the root cause of not taking into account the occurrence of risky situations due to errors in identifying the cause-and-effect relationships of the influence of active factors.

According to these requirements was built operational structural model of the implementation of management actions (Fig. 1). Figure 1 shows: RF – resource flows, $\{Ag_i\}_{i=1}^n$ – process units, EM – executive mechanism, IMS – information and measurement system, CA – command actuator, SU – control system in the ACS, CO – control object.

Signals about the state of the system and the external situation are perceived by the operator through the sensory system and processed in the brain neurostructures in order to form an image of the situation in the target space of the embedded system.

Under emergency conditions in technogenic structures, emergency services and EMERCOM units are located in different points. They have a distributed energy, resource, production and information-management structure, which makes it difficult to make coordinating and guiding decisions especially in emergency modes of operation.

These conditions form requirements to the intelligence and psychophysiological characteristics of operators, as well as to the management apparatus regarding their skills and psychological stability in conditions of risk and emergencies (which result in the emergencies).

4.2 Analysis of the Level of System and Cognitive Risks in Decision-Making in a Complex Technogenic System Under Conditions of Active Factors of Threats and Information Attacks

According to the structural and functional scheme of the information technology of formation of managerial decisions (Fig. 1), we identify classes of managerial actions (which in the case of incorrect decisions can lead to an emergency situation):

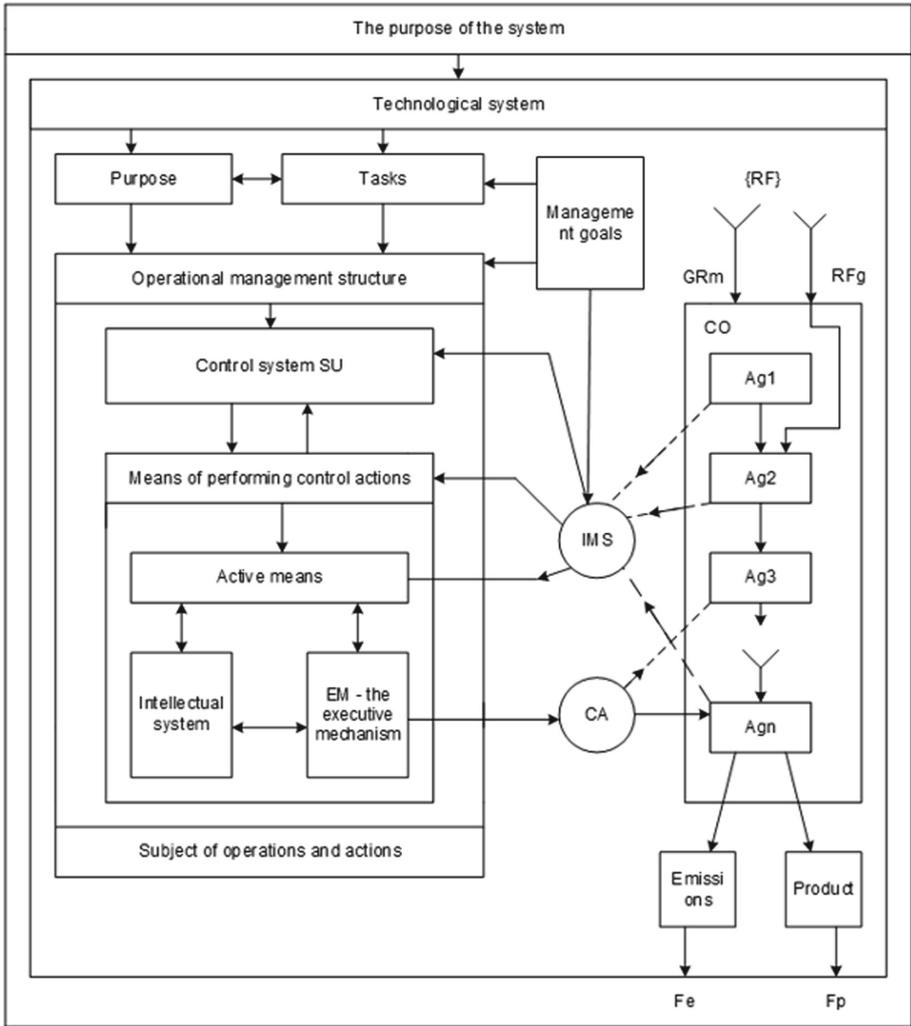


Fig. 1. Operational model of performing control actions in a technogenic system

- information operations (procedures and methods of data processing, statistics, formation of images of situations, methods of estimation of trajectory parameters, classification, data content) and methods of estimation of time characteristics of trajectory changes relative to the target area;
- cognitive operations (planning of scheme of target-oriented tasks solution process, interpretation and prognosis of situations development relatively to the target forming control actions, performed on the basis of dynamic tables according to strategies and control plans on the terminal cycle);

- evaluation of the degree of implementation of goal-oriented management in accordance with risk quality criteria.

For risk assessment we form:

- a set of alternative solutions $A = \{A(U_{ij})\}$;
- set of state trajectories $M(Trak(X_i))$;
- the lines of admissible modes $L(L_m, L_g, L_n)$;
- the line of the emergency mode - L_A ;
- a set of intellectual factors of influence $M[IKFV]$;
- a set of informational threat factors $M[IF_{a,Z}]$;
- set of criteria for risk assessment $M[K_i]$ and alternative partitioning of the space of admissible states.

The dependence of the degree of implementation of management and coordination actions on factors is presented in (Fig. 2).

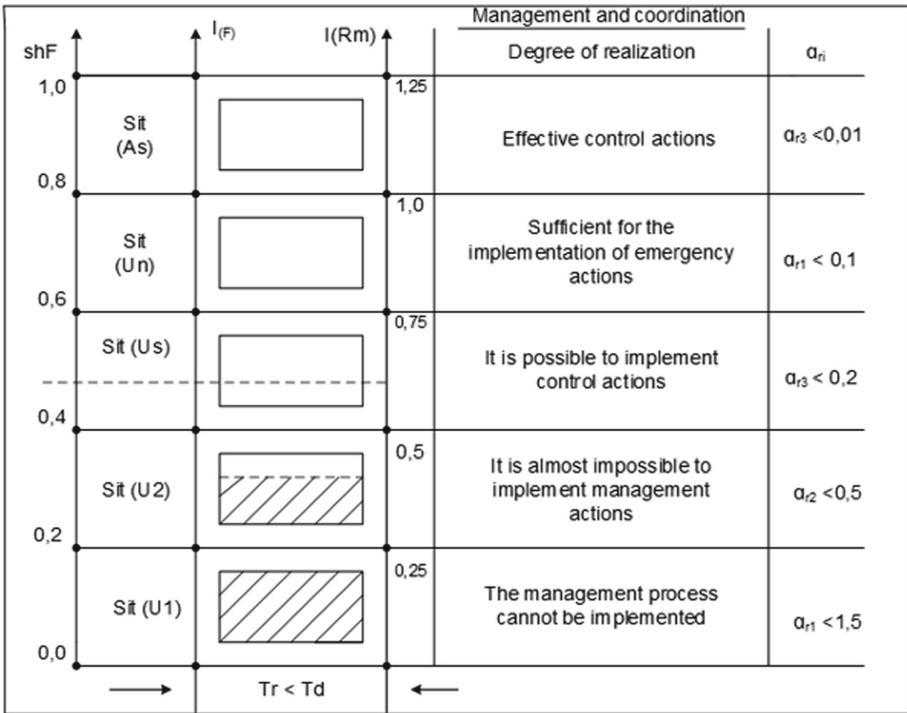


Fig. 2. Possible states and implementation of management actions under the action of active factors as in fuzzy data lead to risk $\left(\alpha_{risk} = \bigcup_i \alpha_{risk?i=1,n}\right)$ - emergency situation

5 Experiment, Results and Discussion

In accordance with the target task and the system load card, a resulting structural diagram of the process of crisis situation formation was developed. Such a crisis situation may arise due to active factors of different nature: informational, systemic, structural, cognitive. Activation of the factors occurs through the performance of action, as a trigger mechanism of causal impacts (AF_i) with a certain level of excitation threshold (Fig. 3)

Accordingly, the system portrait was formed - the scheme of dynamic loading scales (Fig. 4). The developed system portrait demonstrates coordination of loading scales, scales and areas of permissible and emergency states, determination of permissible loading of power-active block unit with permissible state trajectories (R_A), ($trak_i$) and their coordination with the risk scale (α_r) (Fig. 4). Figure 4 shows: Sh_p is the normative load rating scale, Sh_n is the normative scale state, Sh_K is the cognitive load scale, ρ_{h_r} is the risk distribution function, R_A is the active unit power, Sh_r is the accident risk area scale, α_r is the load risk, α_K is the cognitive stress risks.

Accordingly, we build a factor diagram of the influence on the process of risk management and assessment by the operator-cognitive agent (Fig. 5).

Figure 5 shows: KIA is the operator-cognitive agent; $Alg/\alpha(risk)$ is the algorithm for forming the impact on the process; $SitRisk$ is the risk situation with certain factors; L_A , L_m , L_g , L_n is the cognitive factors of the agent; OY is the situation management; $M[trak(X_i/C_i)]$ is the interpretation of the situation by the agent, Fv_i is the management functions.

According to the diagram (Fig. 5), an algorithm for assessing the risks of making incorrect management decisions is constructed (according to the factors that determine the level of errors that can lead to an emergency situation). If one of the components of occupational and cognitive characteristics outweighs the other in terms of significance, then (in certain situations) it may be one of the causes of accidents.

Based on the developed structural scheme of the crisis formation process under the influence of attacks and active threats (Fig. 3), the scheme of dynamic load scales (Fig. 4) and the factor diagram (Fig. 5) was built effective table of management procedures (Table 1). These management procedures must be able to be performed by the operational manager at the maximum load of the energy-intensive facility. The developed table (Table 1) is the basis for creating a systematic and logical structure of tests for the selection of operational personnel.

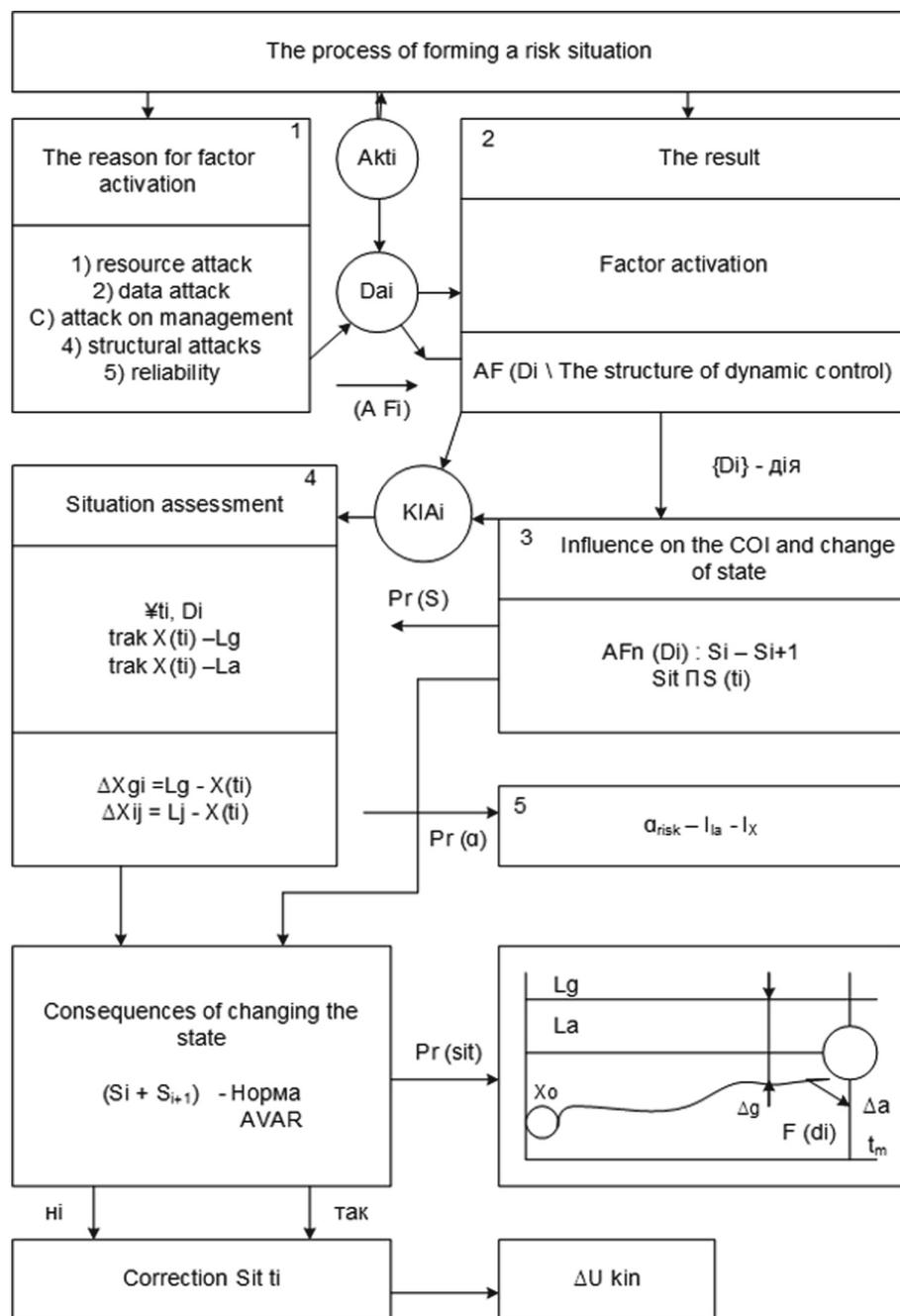


Fig. 3. The process of forming a crisis situation under the action of attacks and active threats (structural scheme)

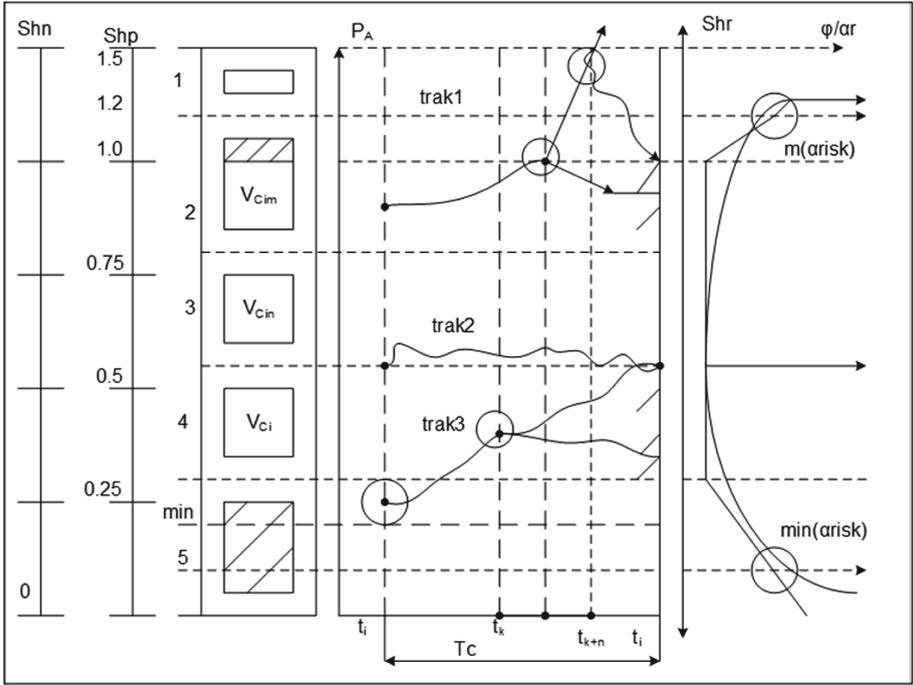


Fig. 4. The process of forming a crisis situation under the action of attacks and active threats (structural scheme)

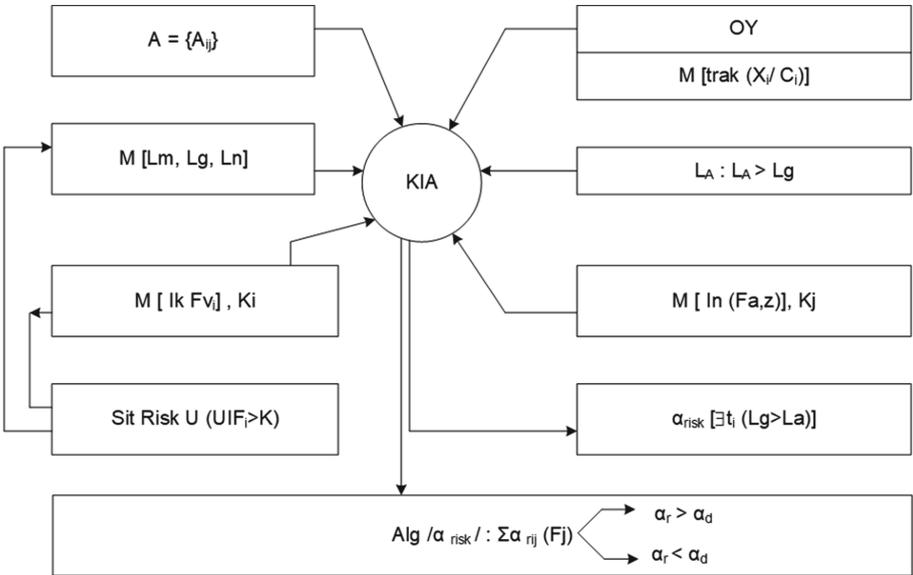


Fig. 5. The process of forming a crisis situation under the action of attacks and active threats (structural scheme)

Table 1. Types of management procedures (which must be able to perform the operator-cognitive agent)

№	Procedure	Code	Coefficient of factor	α_{risk}
1	Situational classification procedure	ΠAD_1	$FKLsit$ (0,5–1,0)	$\leq 0,3$
2	Classification of alternatives	ΠAD_2	F_{ak} (0,7–1,0)	$\leq 0,25$
3	Forming strategies for targeted decisions	ΠAD_3	F_{SR} (0,75–1,0)	$\geq 0,5$
4	Search for emergency management strategy	ΠAD_4	F_{pa} (0,8–1,0)	$\geq 0,8$
5	Forecast of management results	ΠAD_5	F_{rd} (0,5–1,0)	$\geq 0,5$
6	Risk assessments of performed works	ΠAD_6	F_{rvd} (0,5–1,0)	0,75–1,0
7	Risk assessments of non-standard behavior of the operator	ΠAD_7	F_{rnp} (0,5–1,0)	0,3–1,0
8	Choice of strategies with minimal risk by the operator	ΠAD_8	$F_{S(minr)}$ (0,7–1,0)	0,75–1,0
9	Evaluation of the results of anti-crisis decisions	ΠAD_9	F_{AKR} (0,75–1,0)	0,75–1,0
10	The logic of tracing the scheme of decisions	ΠAD_{10}	F_{LR} (0,75–1,0)	$>0,75$

6 Conclusions

In the article the information-functional structure of procedures for selecting and planning management actions in extreme situations was considered. The article presented the development of an information-functional structure of the procedure for selecting and planning management actions in extreme situations.

The conducted systematic and information-cognitive analysis of the process of crisis and conflict situations in the management hierarchy allowed to identify cognitive risk indicators (as a basis for building the characteristics of professional suitability of operational personnel).

The paper substantiates the formulation of the problem of decision-making in the face of threats and information attacks. Based on the given structural scheme of the process of crisis formation under the influence of attacks and active threats, a set of requirements for professional training of personnel was determined (which would ensure effective resolution of crisis situations). The scheme of situational management (based on the operational model of control actions in the man-made

system) was also substantiated and crisis indicators of risks in the formation of control actions (which depend on the cognitive characteristics of the person) were identified.

To form decision-making strategies, the causes of threats and crises (which have information, system, resource, logical components) were analyzed.

The requirements for management procedures based on the system model of dynamic response were formed in the paper. This model includes informational, intellectual and logical components. This ensured the development of an effective strategy for countering threats to resources and information.

The scheme of dynamic scales of loading which are the basis of classifiers of a situation and construction of the scheme of the signaling device of boundary and emergency modes is constructed. This is necessary for informational warning of operational personnel about the impending threat.

References

1. Andonov, S.: Safety Accidents in Risky Industries: Black Swans, Gray Rhinos and Other Adverse Events. CRC Press, 1st edn. (2021). <https://doi.org/10.1201/9781003230298>
2. Balyk, V., Thuong, N.Q.: Statistical synthesis of the principle of rational organization of a complex technical system. In: 2019 International Conference on Engineering and Telecommunication (EnT), pp. 1–4 (2019). <https://doi.org/10.1109/EnT47717.2019.9030569>
3. Bhise, V.: Decision-Making in Energy Systems. CRC Press, 1st edn. (2021). <https://doi.org/10.1201/9781003107514>
4. Boy, G.A.: The Handbook of Human-Machine Interaction: A Human-Centered Design Approach. CRC Press, 1st edn. (2011). <https://doi.org/10.1201/9781315557380>
5. Crozier, R., Ranyard, R., Svenson, O.: Decision Making: Cognitive Models and Explanations. Routledge, 1st edn. (1997). <https://doi.org/10.4324/9780203444399>
6. Groome, D., Brace, N., Edgar, G., et al.: An Introduction to Cognitive Psychology: Processes and Disorders. Routledge, 4th edn. (2021). <https://doi.org/10.4324/9781351020862>
7. Kirwan, B., Ainsworth, L.K.: A Guide To Task Analysis: The Task Analysis Working Group. CRC Press, 1st edn. (1992). <https://doi.org/10.1201/b16826>
8. Kozlova, L.P., Kozlova, O.A.: Using of fuzzy set theory when designing technical systems. In: 2015 XVIII International Conference on Soft Computing and Measurements (SCM), pp. 193–194 (2015). <https://doi.org/10.1109/SCM.2015.7190453>
9. Mutambara, A.: Design and Analysis of Control Systems. CRC Press, 1st edn. (1999). <https://doi.org/10.1201/9781315140940>
10. Sikora, L., Lysa, N., Martysyshyn, R., Miyushkovych, Y.: Information technology for assessing the situation in energy-active facilities by the operator of an automated control system during data sampling. In: Babichev, S., Lytvynenko, V. (eds.) Lecture Notes in Computational Intelligence and Decision Making, pp. 177–187. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-030-82014-5_12

11. Sikora, L., Martsyshyn, R., Miyushkovych, Y., Lysa, N., Yakymchuk, B.: Problems of data perception by operators of energy-active objects under stress. In: *The Experience of Designing and Application of CAD Systems in Microelectronics*, pp. 475–477 (2015). <https://doi.org/10.1109/CADSM.2015.7230909>
12. Sikora, L., Martsyshyn, R., Miyushkovych, Y., Lysa, N., Yakymchuk, B.: Systems approaches of providing the guaranteed functioning of technological structures on the basis of expert coordination of local strategies. In: *2015 Xth International Scientific and Technical Conference “Computer Sciences and Information Technologies” (CSIT)*, pp. 166–168 (2015). <https://doi.org/10.1109/STC-CSIT.2015.7325458>
13. Tattam, D.: *A Short Guide to Operational Risk*. Routledge, 1st edn. (2011). <https://doi.org/10.4324/9781315263649>
14. Watt, R.: *Visual Processing: Computational, Psychophysical, and Cognitive Research: Computational Psychophysical and Cognitive Research*. Psychology Press, 1st edn. (1988). <https://doi.org/10.4324/9781315785080>
15. Wiggins, M.: *Introduction to Human Factors for Organisational Psychologists*. CRC Press, 1st edn. (2022). <https://doi.org/10.1201/9781003229858>
16. Yu, P.: *Multiple-Criteria Decision Making: Concepts Techniques and Extensions*. New-York-London: Plenum Press (1985). <https://doi.org/10.1007/978-1-4684-8395-6>



Expert Decision Support System Modeling in Lifecycle Management of Specialized Software

Yuliia Kordunova^(✉) , Oleksandr Prydatko , Olga Smotr ,
and Roman Golovatyi 

Lviv State University of Life Safety, Lviv, Ukraine
kordunovayulia@gmail.com

Abstract. The paper describes the complexity of the short-term planning process in lifecycle management of specialized software. The critical stages of sprint scope planning in projects, which works by Agile models were explored. Network planning methods have been adapted to determine the critical indicators in lifecycle management of specialized software. An algorithm for automated construction of a network model and its representation in PC memory is proposed, as a data structure. Based on it, algorithms for constructing and traversing graphs of the network model, determining early and late execution time, and determining the critical execution path and time reserves are proposed. Developed algorithms for constructing and traversing graphs of the network model to automate the calculation of its parameters will be embedded in the work of the expert decision support system in lifecycle management of specialized software. The developed expert system will allow making operative decisions on re-planning of duration and the maintenance of project works in real-time. The system was developed using the Java programming language. The results of the system are presented in the development environment IntelliJ IDEA.

Keywords: Software lifecycle management · Agile · Sprint planning · Network graphs

1 Introduction

It is known that the software development process includes many important stages: requirements analysis, planning, design, development and programming, testing, support and operation [10]. One of the crucial stages of development is planning, because at this stage the development team faces several crucial tasks: determining the timing of project development, choosing methods and means of development, establishing methods of implementation and more. If the latter have clear rules and instructions on how to implement a method, a set of practices and means of implementation, the issue of scheduling software development

deadlines is a difficult task and requires a clear analytical calculation. The overall success of the project also depends on the quality and balanced assessment of the amount of work to be performed within the set timeframe.

Nowadays, a number of mathematical and software tools for project planning exist and are successfully used, including software development. However, existing methods are not adapted to a dynamic environment, where in addition to system requirements, the time resources of individual sprints or even the composition of teams may change during development. In such conditions, the application of classical approaches to planning IT projects is experiencing some difficulties.

As a result, there is a need to adapt existing or develop new planning methods for software development that can produce effective results in a dynamic environment. This work is devoted to this issue, namely the adaptation of network planning methods, as well as the development of algorithms for bypassing the graph of the network model.

2 Problem Statement

In general, the process of sprint planning (the next stage of the software life cycle) includes such stages as creating a product backlog, starting a sprint and determining the minimum viable product (MVP), creating a sprint backlog, prioritizing user stories (backlog tasks) and their evaluation. These processes are adapted and tested in traditional project teams working with flexible software development models [18]. These are teams of developers, numbering 5–9 people with the division of responsibilities by roles (positions), who spend most of the sprint's time on solving tasks. The dynamism of such a process depends only on a possible change of scope tasks. Team size and execution time are fixed (Fig. 1)

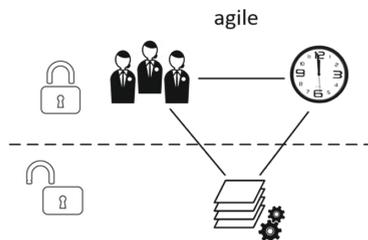


Fig. 1. Limitations on a flexible software lifecycle management model

However, if the focus is made on the teams of developers of operational formations, such as the State Emergency Service of Ukraine, which develops security-oriented services (SOS), the dynamics of specialized software development are characterized not only by the scope of work but also time resource. Some IT departments of operational formations, in addition to general tasks in the field

of computer science and information technology, are engaged in the implementation of applied tasks for informatization of operational and daily activities, as well as design, development and support of computer and software systems security-oriented direction [15].

The members of such development teams in operational formations are the personnel of the relevant services, who, within the scope of their functional responsibilities, combine the activities related to the development of these services with other types of operational or service activities. According to the main participants' specifics of such teams, the dynamics of the project environment take on a slightly different meaning. Dynamics are now characterized not only by the volume of work but also by the time of their implementation. At first glance, we can assume that in such conditions, the process of developing specialized software should be organized according to the cascade model. However, this assumption is wrong, because the development of SOS is characterized by the dynamics of the specification and the need to constantly update the list of works during development. Under these conditions, the development of security-oriented services by project teams of operational (military) formations will acquire the model shown in Fig. 2.

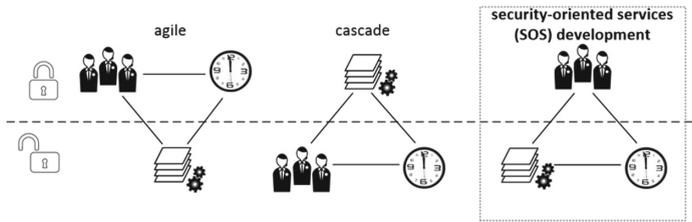


Fig. 2. Flexible and cascading software development models comparison with the actual SOS implementation model

From the presented model we can conclude that the dynamics of planning the volume of work on a particular sprint, as well as the lack of control (limitation) of time for their implementation, may be the cause of untimely or poor quality project implementation. That is why there is a need to study the existing methods of optimal planning of time resources for the development of SOS, which will correspond to the paradigm of flexible management and will allow responding quickly to deviations from the defined work plan. Therefore, the work aims to study existing methods and develop effective mechanisms (algorithms) for resource management of specialized software development projects (security-oriented services) in a dynamic environment (changing the content and scope of work, and adjusting execution time).

The Main Contribution of the Work. According to problem statement, the existing methods and management tools don't correlate with the conditions in which the development of specialized software is carried out, where in addition to variable requirements, execution time is fundamentally important.

So, according to the results of the study of existing methods of software life cycle planning was made a decision to use mathematical methods of planning. Based on it algorithms for constructing and bypassing the graph of the network model were developed for the first time.

These algorithms form the basis of an expert decision support system, which provides an opportunity to increase the efficiency of planning individual stages of software development and make their adjustments in real-time.

3 Literature Review

A large number of Ukrainian and foreign scientists worked on the problems of planning software development projects. In particular, in work [3] it was proved planning takes the center stage in Agile projects and other domains revolve around it. In work [11, 17] the importance of the planning process and expert evaluation of tasks during planning poker has been proved, and the scientific work [8] points out the importance of using the “task board” during planning. The scientific source [14] considered the necessary steps, which should be taken to get the most out of agile software development, and [7] suggested the use of a flowchart tool for decision-making in interdisciplinary research collaboration. The work [4] describes user story estimation based on the complexity decomposition using bayesian networks.

There are also many scientific papers [5, 19, 22] based on the use of network graphs in the project planning process. In particular, the authors in [19] chose the method of solving the planning problem based on the application of the network planning method, which is based on the idea of optimizing the critical path with the involvement of additional limited funds. This approach does not correlate with the development of specialized software. Article [5] describes the process of developing a computer program for solving network optimization problems, but the calculation of the shortest paths is based on Dijkstra’s algorithm, which is not universal and does not work for network planning of the software development process. The article [22] is based on the methods of PERT network planning, the use of elements of graph theory and the method of Gantt charts. This approach is relevant for the cascade management model, which is characterized by consistent execution of tasks and a certain execution time.

Based on papers [1, 2, 12, 20, 21], which lay the basic practices of modern product development planning and product planning practices in dynamic conditions, it was decided to develop flexible algorithms for the project planning process using network graphs in a dynamic environment. The work [20] proposes a planning framework in which multi-fidelity models are used to reduce the discrepancy between the local and global planner. This approach uses high-, medium-, and low-fidelity models to compose a path that captures higher-order dynamics while remaining computationally tractable. In work [21] was demonstrated an informational system or end-to-end workflow on time-series forecasting in Agile processes. Work [2] based on the concept of a minimum viable product in product development, the new concept of a minimal viable production system (MVPS) is

designed. The approach focuses on the reduction of inefficient planning processes due to changing product characteristics and aims to shorten the planning time and the level of maturity for defined planning tasks in order to facilitate early try out of production processes in a series production environment. Authors in work [1] propose a planning assistance platform based on solving the planning problem modeled as a constraint satisfaction problem (CSP). This helps project managers to analyze the project feasibility and to generate useful schedules and charts. In work [12] was argued and developed information graphics technologies of designing models of the processes of multiparameter technical systems in order to increase the effectiveness of determining the influence of many operating parameters on their dynamics, which help us to construct the network planning graph.

Authors in work [13] worked on developing security-oriented services by the hybrid management models. However, the paper describes only the results of the development and doesn't consider the key stages of software life cycle management. The algorithm development issue, in particular the security-oriented systems management, was also addressed by scientists in the work [16]. However, the algorithms developed in this work are difficult to adapt to the life cycle of specialized software management.

4 Materials and Methods

One option for achieving the goal is to use the mathematical apparatus of graph theory and adapt network planning processes by optimizing the calculations of its basic time parameters under the dynamic conditions of the specialized software development.

The adapted method of calculating the parameters of the network model will allow for operational re-planning of the project stages of implementation in a dynamic environment, as well as to determine the critical scope of mandatory work. Network planning will allow tracking in a dynamic environment of the possible risks of overtime, as well as prioritize the phasing of their implementation. Before proceeding to build a network graph, it is important to consider the process of planning design work for software development. Pay special attention to sprint planning. It should be emphasized, that the planning process consists of defining product requirements in the form of user stories [6]. In Agile methodology, it is not acceptable to break down requirements on technical tasks as such an approach does not allow developers to look integrally at the performance of certain program functions.

A user story is a short and simple description of product characteristics from the point of view of a user seeking new opportunities [9]. It includes the full range of development: from design to testing. Such an approach will allow the whole team to participate in the process of evaluating and developing this functionality. Once determined, the user story is broken down into small sub-tasks (functions) that will be performed directly by different team members. From this, it is clear that there is a high probability that several works can be performed in parallel

(which will save a lot of time), and there may be work that depends on the previous ones. That is why determining the critical path that will take the most time to complete is extremely important.

Since works and events are the main elements of a network model in network planning, in the context of Agile scheduling, work is a user story or function that has been shredded. As a unit of measurement use story points are used, which means the relative value, in particular a combination of the development complexity, and the risk associated with it. Each user story or function must have a start and end, which means events on the network graph.

To obtain a network planning graph it is necessary:

- prioritize user stories and functions that need to be performed;
- evaluate user stories and functions in the story points;
- identify previous user stories or functions to be performed for specific user story.

At the beginning of building the network graph, number the initial event $j = 1$, and the previous event ($i = 0$) is absent. From it, draw a vector that will indicate the first user story or function and its weight in the Story Points. If several tasks can be performed in parallel, the required number of vectors is built. Since each user story must end with a final event, mark the event on the graph and increase its sequence number by 1 (do this for all parallel works, increasing j by 1).

If it is needed to build a user story, which is possible only after the previous one end, look for the final event of the work that interests you on the graph, and from it, draw the vector of the desired user story. If the execution of the user story depends on the completion of several previous ones, solve this problem using fictitious work. From each previous user story draw a vector, which means a fictitious work whose weight is 0 s.p., and end them in one final event, which will be the beginning of the new user story you need. Thus, build a network planning graph for the number of user stories and functions that interest you (within the scope of the sprint).

The Algorithm 1 presents the stepwise procedure for constructing a network graph (network model) of a certain amount of work on the sprint. It should be recalled that the main elements of the network model are its events and works, which are evaluated in the story points. This type of estimation allows you to take into account the time and resource constraints of specific software development processes.

The main parameters of the network model for assessment of specialized software development processes at the planning stage include the early time of the event execution T_e the late time of the event execution T_l , event execution reserves R_e , and work performance reserves R_w . The algorithm for determining these parameters, which will be embedded in the work of the expert decision support system, will be discussed below.

Algorithm 1: The algorithm for constructing a network graph (network model) of a certain amount of work on the sprint

Initialization:

```

prevTask[k][p] - array of previous works; pair < i, j >, where i = 1,
j = 2; n- works, n = 1; empty map MapTask < n, pair < i, j >>
for (k = 0; k < prevTask.length; k++) do
    for (p = 0; p < prevTask[k].length; p++) do
        if (prevTask[k][p] == 0) then
            Add n and pair < i, j > to MapTask < n, pair < i, j >>;
            j++; n++;
        end
    end
end
end
i++;
for (k = 0; k < prevTask.length; k++) do
    for (p = 0; p < prevTask[k].length; p++) do
        if (prevTask.length == 1) then
            for (Map.Entry < n, pair < i, j >> works =
                MapTasks.entrySet()) do
                if (prevTask[k][p] == works.getKey()) then
                    for (Map.Entry < i, j > pairSet : pair.entrySet())
                        do
                            i = pairSet.getValue();
                            Add n and pair < i, j > to
                                MapTask < n, pair < i, j >>;
                            j++; n++;
                        end
                    end
                else
                    continue;
                end
            end
        end
        else
            j++;
            for (Map.Entry < n, pair < i, j >> works =
                MapTasks.entrySet()) do
                if (prevTask[k][p] == works.getKey()) then
                    for (Map.Entry < i, j > pairSet : pair.entrySet())
                        do
                            i = pairSet.getValue();
                            Add fict. work 0 and pair < i, j > to
                                MapTask < n, pair < i, j >>;
                            j++;
                        end
                    end
                else
                    continue;
                end
            end
        end
        i = j; j++;
        Add n and pair < i, j > to MapTask < n, pair < i, j >>;
        n++;
    end
end
end
end
Return MapTask < n, pair < i, j >>.

```

In the first stage, the determination of the early time execution of events is implemented. This parameter characterizes the period before which the event can't occur and allows you to control the start of work that is not regulated by the order according to the network model. For the first event $T_{e(1)} = 0$, for all subsequent events of the network model T_e is determined from the expression:

$$T_{e(j)} = T_{e(i)} + t_{(i \rightarrow j)}, \tag{1}$$

where: $T_{e(i)}$ is the early time for the previous event; $t_{(i \rightarrow j)}$ is the amount of work (unified resource) that precedes the event j (takes into account time and human constraints in Story Point).

It is necessary to provide the possibility of correct calculation of the early time execution for those events that are preceded by several works, taking into account the requirement $T_{e(i)} \rightarrow max$:

$$T_{e(j)} = max \begin{cases} T_{e(k)} + t_{(k \rightarrow j)} \\ \dots \\ T_{e(m)} + t_{(m \rightarrow j)}, \end{cases} \tag{2}$$

where k, m are the indices of events preceding the event j .

Algorithmically, the procedure for determining the early term of work with all constraints can be represented as an algorithm for traversing the graph of the network model with an iterative definition of the parameter $T_{e(j)}$ (Algorithm 2), which was developed by the authors for the first time and adapted for flexible project management.

The term of early execution of events for the last event of the network model n is the term of execution of the whole complex of sprint's works. Next, to control the terms by which the events of the network model must be fully completed, to avoid cases of increasing time for the implementation of the entire project, it is necessary to determine the late time of event execution $T_{l(i)}$:

$$T_{l(i)} = T_{l(j)} + t_{(i \rightarrow j)}, \tag{3}$$

where: $T_{l(j)}$ is the late time of event execution; $t_{(i \rightarrow j)}$ is the amount of work performed after the event i and before event j (taking into account time and human constraints in Story Point).

The late time of event execution T_l for the last event of the network model is equal to the early time of the event execution T_e :

$$T_{l(nlast)} = T_{e(nlast)} \tag{4}$$

Similar to the previous cases, when the next event i initialize the execution of $n \geq 2$ number of works, the late time of the event i is taken as follows:

$$T_{l(i)} = min \begin{cases} T_{l(k)} + t_{(i \rightarrow k)} \\ \dots \\ T_{l(m)} + t_{(i \rightarrow m)}, \end{cases} \tag{5}$$

where k, m are the indices of derived events, from the event i .

Algorithm 2. The algorithm for traversing the graph of the network model to determine the indicator $T_{e(j)}$

Initialization:

i - index of the previous event, $i = 0$;

j - index of the next event, $j = 1$;

n - the volume of network model events;

$\{t_{i \rightarrow j}\}$ - set of network model works;

$\forall j = 1$ then $T_{e(1)} = 0$;

$i ++$;

for ($j = i + 1$; $j \leq n$; $j ++$) **do**

if ($i \rightarrow j \in \{t_{i \rightarrow j}\}$) **then**

$T_{e(j)} = T_{e(i)} + \{t_{i \rightarrow j}\}$;

if ($T_{e(j)} \in \{T_{e_j}\}$) **then**

$max(\{T_{e_j}\}; \{T_{e(j)}\})$;

if ($\{T_{e_j}\} < T_{e(j)}$) **then**

$T_{prev(j)} \leftarrow T_i$;

$HashMap < T_j, T_{prev(j)} > \leftarrow < T_j, T_{prev(j)} >$;

$HashSet\{T_{e_j}\} \leftarrow T_{e(j)}$ >;

else

 | continue *for*;

end

else

$T_{prev(j)} = T_i$;

$HashMap < T_j, T_{prev(j)} > \leftarrow < T_j, T_{prev(j)} >$;

$HashSet\{T_{e_j}\} \leftarrow T_{e(j)}$;

 continue *for*;

end

else

 | continue *for*;

end

end

$i ++$;

if ($i \neq n$) **then**

 | start *for*;

end

Return:

$\{T_{e_j}\}$ - the set of early term execution of n event of the network model;

$HashMap < T_j, T_{prev(j)} >$ - the set of previous events for the next events in the format $< Key, Value >$.

T_l for the first event of the network model, provided the correct execution of preliminary calculations, will be equal to the early time execution of this event T_e :

$$T_{l(nfirst)} = T_{e(nfirst)} \quad (6)$$

For all network model events, except the first and last, the following inequalities must be satisfied:

$$T_{l(n)} \geq T_{e(n)} \quad (7)$$

Algorithmically, the procedure for determining the late deadline for work, taking into account all the limitations, can be represented as an algorithm for traversing the graph of the network model with the iterative determination of the parameter $T_{l(i)}$ (Algorithm 3). The input data for the operation of the algorithm is the result of traversing the graph according to the previous algorithm.

Algorithm 3. The algorithm for traversing the graph of the network model to determine the indicator T_{l_i}

Initialization:

```

i - index of the previous event, i = 0;
j - index of the next event, j = n;
n - the volume of network model events;
{ti→j} - set of network model works;
{Tej} - set of early time execution events;
∀j = n then  $T_{l(n)} = T_{e(n)}$  ;
for (i = j - 1; i ≥ 1; i --) do
  | if (i → j ∈ {ti→j}) then
  |   |  $T_{l(i)} = T_{l(j)} - \{t_{i \rightarrow j}\};$ 
  |   | if ( $T_{l(i)} \in \{T_{l_i}\}$ ) then
  |   |   |  $\min(\{T_{l_i}\}; \{T_{l(i)}\});$ 
  |   |   | if ( $\{T_{l_i}\} < T_{l(i)}$ ) then
  |   |   |   |  $HashSet\{T_{l_i}\} \leftarrow T_{l(i)};$ 
  |   |   |   | else
  |   |   |   |   | continue for;
  |   |   |   |   | end
  |   |   | else
  |   |   |   |  $HashSet\{T_{l_i}\} \leftarrow T_{l(i)};$ 
  |   |   |   |   | continue for;
  |   |   |   | end
  |   | else
  |   |   | continue for;
  |   | end
  | end
  | j --;
  | if (j ≠ 1) then
  |   | start for;
  |   | end
Return:
{Tli} - the set of late term execution of n event of the network model;

```

The results of the calculations allow us to estimate the maximum amount of work given the available resources needed to complete the sprint. The maximum

amount of work in the network model reflects the critical path. The definition of the critical path is organized by the network of works $t_{(i \rightarrow j)}$ from the next late event to the event preceding it (the previous event of the vertex of the network graph specified in the corresponding sector). The construction of the critical path begins with the final event T_n and ends with the first event T_1 :

$$T_n \rightarrow [t_{(j \rightarrow n)}] \rightarrow T_j \rightarrow [t_{(k \rightarrow j)}] \rightarrow T_k \rightarrow \dots \rightarrow T_m \rightarrow [t_{(i \rightarrow m)}] \rightarrow T_i \rightarrow [t_{(l \rightarrow i)}] \rightarrow T_1, \quad (8)$$

where: T_n is the last event of the network model (the first for the critical path); T_j is the late intermediate event of the network model; T_i is the early intermediate event of the network model; T_k, T_m are the intermediate events of the network model; T_1 is the first event of the network model (the last for the critical path).

Algorithmically, the procedure for determining the critical path based on the results of traversing the graph of the network model is shown in Algorithm 4. The input data for this procedure are the results of previous models.

Algorithm 4. The algorithm for traversing the graph of the network model to determine the critical path

Initialization:

j - index of the event, $j = n$;

k - temporary variable;

$HashMap < T_j, T_{prev} >$ - the set of previous events for the next events T_j in the format $< Key, Value >$.

$Stack.add(T_j)$;

while ($j \geq 1$) **do**

$Stack.add(k)$;

$j - -$;

end

Return:

$Stack$.

The peculiarity of the critical path of the execution of work is that it does not contain any resource reserve to achieve events (early and late terms of execution for all events at the critical level). This indicates that any delay in the implementation of network model events that lie within the critical path will encourage incomplete execution of a certain amount of work on the sprint or exceed the allowable time of their implementation.

However, other works which are included in the network model and do not belong to the critical path may have some reserves to achieve events and perform work. The value of these reserves will allow controlling the limits of the critical time of start and end of work that is not within the critical path. The calculation of these resource reserves is performed using the next expressions:

$$R_e(j) = T_{l(j)} - T_{e(j)} \quad (9)$$

$$R_{pw}(t_{(i \rightarrow j)}) = T_{l(j)} - T_{e(i)} - t_{(i \rightarrow j)} \quad (10)$$

$$R_{fw}(t_{(i \rightarrow j)}) = T_{e(j)} - T_{e(i)} - t_{(i \rightarrow j)} \tag{11}$$

$$R_{dw}(t_{(i \rightarrow j)}) = T_{e(j)} - T_{l(i)} - t_{(i \rightarrow j)}, \tag{12}$$

where: $R_e(j)$ is the event time reserve; $R_{dw}(t_{(i \rightarrow j)})$ is the independent work reserve; $R_{pw}(t_{(i \rightarrow j)})$ is the full work reserve; $R_{fw}(t_{(i \rightarrow j)})$ is the free work reserve; $T_{l(i)}$ is the late time of the previous event; $T_{l(j)}$ is the late time of the next event; $T_{e(i)}$ is the early time of the previous event; $T_{e(j)}$ is the early time of the next event; t is the scope of work; j is the next event of the network model; i is the previous event of the network model.

The Algorithm 5 shows the stepwise procedure for traversing the graph of the network model to determine these reserves. It should be emphasized that event time reserve $R_e(j)$ characterizes the maximum allowable period for which it is possible to delay the execution of event n without increasing the critical path and resource constraint on the sprint. The full reserve of work $R_{pw}(t_{(i \rightarrow j)})$ describes how long it is possible to postpone the start of work $t_{(i \rightarrow j)}$ or increase its duration within available resources and without increasing the total sprint time (critical path length). The free reserve of work $R_{fw}(t_{(i \rightarrow j)})$ is the time for which it is possible to postpone the start of execution of work $t_{(i \rightarrow j)}$ or extend its duration without violating the early term of execution events in the network model. The independent work reserve $R_{dw}(t_{(i \rightarrow j)})$ characterizes the delay time of the start of work $t_{(i \rightarrow j)}$ without increasing the total term of the sprint’s tasks complex and without delaying any of the other works of the network model.

5 Experiment, Results and Discussion

The algorithms for automated construction and determination of network model parameters, when planning project works, were tested during the development of a security-oriented system commissioned by the State Emergency Service of Ukraine “Development consulting assistance software system to the population in case of threat or occurrence of emergencies with an integrated notification function based on mobile platforms”. Input data and tasks for development were provided by the State Emergency Service of Ukraine during martial law, which confirms the work in dynamic conditions and limited time resources. After receiving the task, the project backlog was formed and determined MVP by the Product Owner. In order to determine the date of the first release, the team estimated user stories and shredded features in Story Points and determined the relationships between user stories, and the priority of execution. As a result, Table 1 was formed.

Table 1. A set of input data for building a network graph

Number of task	1	2	3	4	5	6	7	8	9	10	11
Story point	27	35	15	12	25	30	17	13	15	27	35
Number of previous task	0	0	1	1	2	3,4	5	5	5	7,8,9	6,10

By implementing the algorithm of constructing a network graph (network model) of a certain amount of work on the sprint, the procedure of constructing a network graph in the IntelliJ IDEA development environment by the Java programming language was implemented.

Figure 3 shows the result of building a network model of project works identified in the sprint.

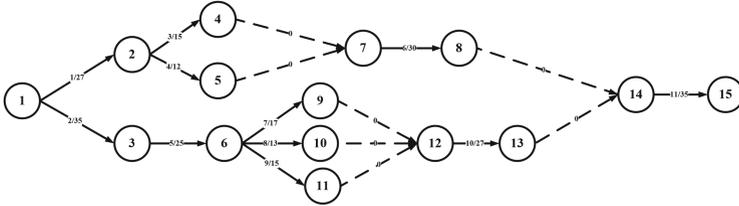


Fig. 3. The network model of the software development process, presented in the graph form

To properly characterize all software development processes at the planning stage, it is necessary to determine all the parameters of the sprint’s network model. Initially, graph traversal algorithms were implemented to determine the early and late execution times of sprint (project) events. On the basis of Algorithms 2 and 3 the iterative process of specified parameters calculation is programmatically realized.

Based on the results of calculating the early and late time execution events, the critical path is programmatically determined (according to Algorithm 4). In Fig. 4 the result of critical path automated determination of the studied network model is presented. The obtained result indicates events that are critical and the implementation of which does not involve time reserves.

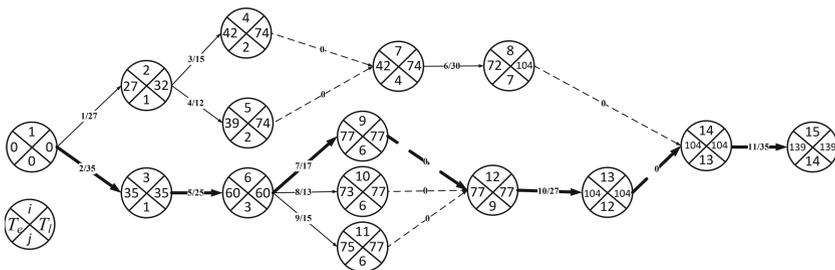


Fig. 4. The result of the critical path, the early and late execution times of events calculation of the software development

Algorithm 5. The algorithm for traversing the network model graph to determine time reserves

Initialization:

i - index of the previous event, $i = 0$;
 j - index of the next event, $j = n$;
 n - the volume of network model events;
 $\{t_{i \rightarrow j}\}$ - set of network model works;
 $\{T_{e_j}\}$ - set of the early terms executed events;
 $\{T_{l_j}\}$ - set of the late terms executed events;
 $Stack$ - the critical path of network model;

while ($j \geq 1$) **do**

$R_{e(j)} = T_{l(j)} - T_{e(j)}$;

if ($R_{e(j)} == 0$) **then**

$j --$;

 continue *while*;

else

$HashSet\{R_j\} \rightarrow R_{e(j)}$;

$j --$;

 continue *while*;

end

end

$i \rightarrow n$;

for ($i = j - 1; i \geq 1; i --$) **do**

if ($i \rightarrow j \in \{t_{i \rightarrow j}\}$) **then**

if ($T_i \notin Stack$) **then**

$R_{pw}(t_{i \rightarrow j}) = T_{l(j)} - T_{e(i)} - t_{i \rightarrow j}$;

$R_{fw}(t_{i \rightarrow j}) = T_{e(j)} - T_{e(i)} - t_{i \rightarrow j}$;

$R_{dw}(t_{i \rightarrow j}) = T_{e(j)} - T_{l(i)} - t_{i \rightarrow j}$;

$HashSet\{R_{pw}(t_{i \rightarrow j})\} \rightarrow R_{pw}(t_{i \rightarrow j})$;

$HashSet\{R_{fw}(t_{i \rightarrow j})\} \rightarrow R_{fw}(t_{i \rightarrow j})$;

$HashSet\{R_{dw}(t_{i \rightarrow j})\} \rightarrow R_{dw}(t_{i \rightarrow j})$;

 continue *for*;

else

 continue *for*;

end

else

 continue *for*;

end

end

$j --$;

if ($j \neq 1$) **then**

 start *for*;

else

Return:

$\{R_{e(j)}\}, \{R_{pw}(t_{i \rightarrow j})\}, \{R_{fw}(t_{i \rightarrow j})\}, \{R_{dw}(t_{i \rightarrow j})\}$.

end

The last step is to determine the event reserves using time reserve algorithm (Algorithm 5). Table 2 shows the result of the specified algorithm of the software implementation and its application to determine time reserves for the studied network model.

Table 2. The result of a program that calculates time reserves

<i>Event</i>	2	4	5	7	8	10	11
<i>R_e</i>	32	32	35	32	32	4	2

Based on the results, we can conclude that the development period of the MVP of this software is 139 Story Points. The critical path is the path that passes through events 1, 3, 6, 9, 12, 13, 14, and 15. Events 2, 4, 5, 7, 8, 10, and 11 have time reserves. If you need to make changes to the list of works or adjust the time to perform a certain amount of work, the developed system for estimating the parameters of the network model based on the developed algorithms will instantly re-evaluate the main characteristics and anticipate possible deviations from the plan. This subsystem forms the basis of the decision support system for the operational management of the developing specialized software process in a dynamic environment. After all, it will allow you to monitor the critical path of work, available time reserves, and the amount of unfinished work in real-time.

6 Conclusions

In dynamic conditions, the planning stage of specific software development processes is extremely important. As the experience of such development has shown, the existing methods and management tools do not correlate with the conditions in which the development of specialized software is carried out, where in addition to variable requirements, execution time is fundamentally important. According to the results of the study of existing methods of software life cycle planning in scientific work, the following results were obtained:

Based on the results of optimization of mathematical methods of network planning for life cycle management processes of specific software, algorithms for constructing and bypassing the graph of the network model are developed to determine its main parameters, which provide an opportunity to increase the efficiency of planning individual stages of software development and make their adjustments in real time.

Based on the scientific results and implemented algorithms, an expert computer system was obtained, which allows determining the basic parameters of the network model and is used to support operative decision-making in the process of short-term life cycle planning of specialized software development projects.

References

1. Belkasmı, M.G., et al.: Global IT project management: an agile planning assistance. In: *Advances in Smart Technologies Applications and Case Studies*, pp. 575–582. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-53187-4_63
2. Bertling, M., Caroli, H., Dannapfel, M., Burggraf, P.: The minimal viable production system (MVPS) – an approach for agile (automotive) factory planning in a disruptive environment. In: *Advances in Production Research*, pp. 24–33. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-03451-1_3
3. Boral, S.: Domain V: Adaptive Planning, December 2016. https://doi.org/10.1007/978-1-4842-2526-4_6
4. Durán, M., Juárez-Ramírez, R., Jiménez, S., Tona, C.: User story estimation based on the complexity decomposition using Bayesian networks. *Programm. Comput. Softw.* **46**, 569–583 (2020). <https://doi.org/10.1134/S0361768820080095>
5. Dymova, H., Larchenko, O.: Development of a computer program for solving network optimization problems. *Comput.-Integr. Technol. Educ. Sci. Prod.* **41**, 143–151 (2020). <https://doi.org/10.36910/6775-2524-0560-2020-41-23>
6. Hallmann, D.: “I don’t understand!”: toward a model to evaluate the role of user story quality. In: Stray, V., Hoda, R., Paasivaara, M., Kruchten, P. (eds.) *Agile Processes in Software Engineering and Extreme Programming*, pp. 103–112. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-49392-9_7
7. Jansen, U., Schulz, W.: Flowchart tool for decision making in interdisciplinary research cooperation. In: *International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management*, pp. 259–269. Springer (2017). https://doi.org/10.1007/978-3-319-58466-9_24
8. Karras, O., Klünder, J., Schneider, K.: Is task board customization beneficial? In: *International Conference on Product-Focused Software Process Improvement*, pp. 3–18. Springer (2017). https://doi.org/10.1007/978-3-319-69926-4_1
9. Khmel, M., Prydatko, O., Popovych, V., Tkachenko, T., Kovalchuk, V.: Students r&d projects as a tool for achieving program competencies. *Information and communication technologies in modern education: experience, problems, prospects ...*(2021)
10. Kordunova, Y., Smotr, O., Kokotko, I., Malets, R.: Analysis of the traditional and flexible approaches to creating software in dynamic conditions. *Manage. Dev. Complex Syst.* 71–77 (2021). <https://doi.org/10.32347/2412-9933.2021.47.71-77>
11. Lenarduzzi, V., Lunesu, M.I., Matta, M., Taibi, D.: Functional size measures and effort estimation in agile development: a replicated study, vol. 212, May 2015. https://doi.org/10.1007/978-3-319-18612-2_9
12. Liaskovska, S., Martyn, Y., Malets, I., Prydatko, O.: Information technology of process modeling in the multiparameter systems, pp. 177–182, Aug 2018. <https://doi.org/10.1109/DSMP.2018.8478498>
13. Martyn, Y., Smotr, O., Burak, N., Prydatko, O., Malets, I.: Software for shelter’s fire safety and comfort levels evaluation. In: *Communications in Computer and Information Science*, pp. 457–469. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-61656-4_31
14. Meier, A., Ivarsson, J.C.: Agile software development and service science. *GSTF J. Comput. (JoC)* **3**(3), 1–5 (2013). <https://doi.org/10.7603/s40601-013-0029-6>

15. Prydatko, O., Kordunova, Y., Kokotko, I., Golovatiy, R.: Substantiation of the managing student R&D projects methodology (on the example of the educational program computer science). *Information and communication technologies in modern education: experience, problems, prospects ...* (2021)
16. Prydatko, O., Popovych, V., Malets, I., Solotvinskyi, I.: Algorithm of rescue units logistic support planning in the process of regional life safety systems development. In: *MATEC Web of Conferences*, vol. 294, p. 04002, January 2019. <https://doi.org/10.1051/matecconf/201929404002>
17. Sachdeva, V.: Requirements prioritization in agile: use of planning poker for maximizing return on investment. In: *Advances in Intelligent Systems and Computing*, pp. 403–409. Springer International Publishing, July 2017. https://doi.org/10.1007/978-3-319-54978-1_53
18. Schwaber, K., Sutherland, J.: *The scrum guide*. Scrum Alliance (2020)
19. Seidykh, O., Chobanu, V.: Optomozation of the network graphics of the complex of works. *Mod. Eng. Innov. Technol.* **1**(3), 61–67 (2018). <https://doi.org/10.30890/2567-5273.2018-03-01-009>
20. Tordesillas, J., Lopez, B.T., Carter, J., Ware, J., How, J.P.: Real-time planning with multi-fidelity models for agile flights in unknown environments, pp. 725–731 (2019). <https://doi.org/10.1109/ICRA.2019.8794248>
21. Uzun, I., Lobachev, I., Gall, L., Kharchenko, V.: Agile Architectural Model for Development of Time-Series Forecasting as a Service Applications, pp. 128–147. Springer International Publishing, July 2021. https://doi.org/10.1007/978-3-030-82014-5_9
22. Velykodniy, S., Burlachenko, Z., Zaitseva-Velykodna, S.: Architecture development of software for managing network planning of software project reengineering. *Innov. Technol. Scient. Sol. Ind.* **2**(8), 25–35 (2019). <https://doi.org/10.30837/2522-9818.2019.8.025>

Data Engineering, Computational Intelligence and Inductive Modeling



Machine Learning of the Biotechnic System for Gastroesophageal Reflux Disease Monitoring

Vsevolod Novikov¹ , Mariia Voronenko¹ , Anastasiia Novikova² , Oleg Boskin¹ , Oleksii Tyshchenko³ , Yuriy Rozov¹ , Yuriy Bardachov¹ , and Svitlana Vyshemyrska¹ 

¹ Kherson National Technical University, Kherson, Ukraine

vsevolodnovikov@live.com, mary_voronenko@i.ua,

{rozov.yuriy,Y.Bardachov,vyshemyrska.svitlana}@kntu.net.ua

² Kherson State Maritime Academy, Kherson, Ukraine

³ Institute for Research and Applications of Fuzzy Modeling, CE IT4Innovations, University of Ostrava, Ostrava, Czech Republic

Abstract. The article is devoted to the study of gastroesophageal reflux disease development. The main research contribution is that the study implements prognostic, morpho-functional models to automate the differential diagnostics process. Also, the research developed a special methodology for automating the differential diagnostics process using artificial neural networks based on predictive morpho-functional models. The system analysis method was applied. This method allows you to study analyzed problems and disease at various systems organization levels, including macro and micro levels to highlight the characteristics, symptoms, syndromes, and signs necessary for private diagnosis, and in the study, the use of algorithms for evaluating the results dispersion was further developed, which made it possible to assess the informativeness of signs about the corresponding nosological disease form. The methods and techniques for treating the disease were analyzed. A faster and more reliable method was proposed for monitoring the food effect on the gastroesophageal reflux disease reaction. Statistical processing of the research results is carried out. The reliability of the data is shown. For a more reliable further diagnosis, machine learning of the biotechnical disease monitoring system was carried out. The machine is properly trained and classifies the image. Regression analysis showed the model reliability built using machine learning. After conducting experiments and subsequent analysis of the results, we obtained an accuracy of 99%. The system has correctly learned to classify data. Regression analysis showed an almost linear regression.

Keywords: Monitoring · Gastroesophageal reflux disease · Heart rate · Machine learning

1 Introduction

Gastroesophageal reflux disease is a disease of the esophagus, which is caused by the reflux of gastric contents into the esophagus. The reasons for this phenomenon are the pathological work of the diaphragm: thinning or underdevelopment of muscle tissue, which is combined with high intra-abdominal pressure, all this leads to a displacement of the abdominal organs into the chest, which in turn can lead to hiatal hernia [5].

The main symptomatology is heartburn, which occurs after eating or taking the body into a horizontal position. The secondary symptoms are regurgitation, belching, cardiac syndrome, orography, bronchopulmonary syndrome, and anemic syndrome.

Gastrointestinal tract diseases are among the most common pathologies in the world. Among such pathologies, Gastroesophageal reflux disease (GERD) gets more development. So, according to scientists' data from Northwestern University of the United States [18] for 2015, US spending on the GERD amounted to about \$ 20 billion. According to the American College of Gastroenterology data [34], heartburn affects about 60 million Americans at least once a month.

As a treatment for this pathology, combined methods of drug therapy with dietary nutrition, or endoscopic methods such as the Stretta procedure [14] and transoral non-surgical fundoplication [8], or surgical interventions are used. If we take traditional methods of treatment, they all include dietary nutrition, but often there is a problem here that in addition to the traditional groups of products provoking GERD (chocolate, alcohol, coffee, tomatoes, soda) there's individual susceptibility to the specific products, so we can see that in different recommendations have mutually exclusive instructions, for example [32,33] eating broccoli, spinach or pasta or tea.

Besides, many publications note that exacerbation of reflux is caused by a specific product only in a certain number of people, for example, by cucumber. Also consider that GERD is usually accompanied by other gastrointestinal diseases such as gastritis, duodenitis, and others, respectively, they will also require a change in diet. Accordingly, each person suffering from reflux must by himself monitor the reaction of his body to a particular product, but it is quite difficult to determine from which product a relapse occurs, since symptoms may worsen gradually.

The main contribution is as follows:

1. the study implements prognostic, morpho-functional models to automate the process of differential diagnostics,
2. the study developed a special methodology for automating the process of differential diagnostics using artificial neural networks based on predictive morpho-functional models,
3. the system analysis method was applied. This method allows you to study the body, allows you to study the analyzed problem and disease at various levels of systems organization, including macro and micro levels to highlight the characteristics, symptoms, syndromes, and signs necessary for private diagnosis, and

4. in the study, the use of algorithms for evaluating the dispersion of results was further developed, which made it possible to assess the informativeness of signs about the corresponding nosological form of the disease.

This area is poorly understood, so the topic we have chosen is relevant.

The work consists of the following sections. Section 2 defines the tasks that we will solve in our research. Also in this section, we discuss the general formulation of the solution to our problem. Section 3 provides a review of the literature on existing methods of differential diagnosis. In Sect. 4, we describe the materials and methods used in our study, as well as the sequence of experiments. In Sect. 5, we run experiments and get results. Section 6 presents an analysis of the research results. Section 7 summarizes and concludes.

2 Review of Literature

There are a large number of processes in the body that can be registered or, in medical terms, diagnosed. The main methods [4, 20, 21] for the diagnosis of gastroesophageal reflux disease are:

- gastroscopy with the examination of the esophagus, stomach, and duodenum [29, 31];
- X-ray examination of the upper gastrointestinal tract with barium [2, 17, 28];
- manometry of the esophagus [12, 23, 27];
- pH-metry is also used for diagnostics [9, 13, 19, 24].

Today, a promising direction in the study of pH-impedance measurement by the impedance jump method is widely used [3]. Studies at the University Children’s Hospital in Belgrade [16] have shown that the new pH-MII monitoring method has a higher detection rate for gastroesophageal reflux disease compared to the single pH meter method. In the work “New Diagnostic Tests for GERD” [11], the authors show that new methods are needed in the diagnosis of GERD; as a non-invasive method, they propose measuring the level of pepsin in saliva [1].

The work “Ambulatory Monitoring for Reflux” [22] showed the effectiveness of monitoring the pH of the esophagus and proposed new indicators for assessing the course of the disease, based on tissue permeability and bolus clearance.

It is also important to note the degree of manifestation of secondary symptoms. The fact is that according to the study by I. Maev, A. Kazyulin, G. Yurenev [30], cardiac syndrome manifests itself in 78.8% of patients. Accordingly, his symptom tachycardia can be an indicator in our study.

The same study [30] states that cardiac syndrome with reflux may be associated with the development of reflex angina on the background of coronary atherosclerosis or intact coronary arteries. This, too, will be taken into account in our study.

However, despite advances in this subject area, the problem of assessing the consequences of reflux disease on the human body has not yet been resolved. The solution to this problem can be achieved through the use of modern methods, which are currently successfully applied in various fields of scientific research [6, 7].

3 Problem Statement

The goal of this study is to determine a fast and reliable method for monitoring the effect of food on the response of gastroesophageal reflux disease, as well as to perform machine learning of a biotechnical monitoring system for this disease.

To achieve this goal, the following tasks were solved:

- developed a special technique for automating the process of differential diagnosis using artificial neural networks based on predictive morphofunctional models;
- the method of system analysis was applied to highlight the characteristics, symptoms, syndromes, and signs necessary for private diagnostics;
- a faster and more reliable method for monitoring the effect of food on the reaction of gastroesophageal reflux disease has been proposed;
- machine-learning of the biotechnical disease monitoring system was carried out;
- the reliability of the model built using machine learning was tested, and the accuracy obtained was 99%.

The work aims to test the possibility of using a non-invasive method of monitoring bpm. It is proposed to use as a monitoring food sensitivity or even intolerance to a certain type of food in the whole gastrointestinal tract. To confirm our hypothesis, we will use Student's t-test and Microsoft Azure machine learning technology.

4 Material and Methods

4.1 Data

The study involved 30 volunteers. The first group consisted of 15 people who were diagnosed with gastroesophageal reflux disease of varying severity. At the start of the study, all participants in this group were in remission. The second group included 15 volunteers, whose health condition can be described as conditionally healthy persons. To confirm this status, the volunteers were examined by a gastroenterologist (however, without an esophagogastroduodenoscopy study). Also the volunteers passed a complete blood count with a formula, blood tests for IgA, IgG, and an analysis for IgM antibodies to *Helicobacter pylori*.

The table below represents the data that show the features of the study group (Table 1). There are factors [5] that affect the development of this disease, course of GERD [15].

4.2 Experimental Technique

The measurements were carried out during 30 days at the time of remission on 15 volunteers and 15 volunteers of the control conditionally healthy group. During the study, all volunteers had to eat and get up/go to bed at about the same time:

Table 1. Data of test volunteers

Designation of the node	Affected percentage of volunteers
Age (min-max)	18–76
Gender:	
Male	57,6
Female	42,4
Factors that can affect the development of pathology:	
Helicobacter pylori infection	0
Tobacco smoking	34,6
Obesity	15,6
High-fat diet	42,3
High alcohol consumption	0,5

6:00 - to wake up, 6:30 - a breakfast, 12:00 - a lunch, 18:00 - a dinner, 20:00 - go to sleep. During the study, volunteers participating in the study adhered to one menu.

The analysis of the research part is carried out with the use of systems analysis, with selected subsystems of the digestive system, as well as their elements at the chemical level. Building a mathematical model, we define some parameters. Some physiological systems take part in digestion, we will present them as a train A :

$$A = \{a_1, a_2, \dots, a_i\}, \quad (1)$$

where a_i are the functional systems of the body involved in digestion (mouth, stomach, etc.), $i = 1, \dots, n$.

Each functional system is represented by a number of subsystems (glands), so we present them as a tuple B :

$$B = \{b_1, b_2, \dots, b_j\}, \quad (2)$$

where $j = 1, \dots, m$ is the number of subsystems of functional digestive systems.

Each such system either produces or absorbs a number of chemical elements. This series is a set C :

$$C = \{c_1, c_2, \dots, c_q\}, \quad (3)$$

where $q = 1, \dots, l$.

This set of elements consists of those that entered the body (c_x) and those elements that coincided (c_y) with the body, i.e.:

$$c_q = c_x + c_y. \quad (4)$$

To test the operation of all digestive systems, the conjuncture function must be used in the simulation:

$$R = \prod a_{(\eta-\gamma) \cdot c_q/b_j} \quad (5)$$

where η is the percentage of chemical elements remaining in the subsystem, γ is the percentage of chemical elements that have left the subsystem; R is functional that describes the digestion work.

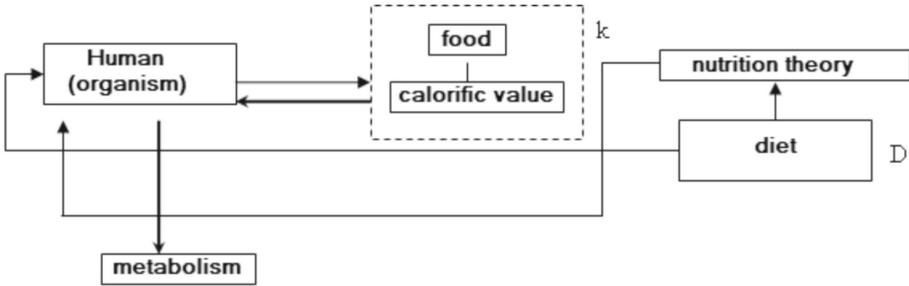


Fig. 1. Nutrition theory analysis

Among the issues that need to be considered in the analysis of the physiology of nutrition, we can identify three main issues (Fig. 1), which usually affect the person himself. Model (5) will change its appearance due to the appearance of components that reflect the diet and caloric content of food, i.e.:

$$R = kd_n \prod a_{(\eta-\gamma) \cdot c_q / b_j} \tag{6}$$

where n takes into account the diet.

It is also necessary to take into account the amount of ballast substances and nutrients. It is necessary to introduce some component that will indicate the normal functioning of the digestive system as a whole. On the other hand, the components of model (7) c_q will consist of the sum of nutrients (nu) and ballast substances (bth).

Given that we evaluate the work of the entire digestive system, it is necessary to introduce a quantifier of commonality. Thus, Formula (7) will take the form:

$$R = kd_n \forall \prod a_{(\eta-\gamma) \cdot c_q / b_j} = Z \tag{7}$$

where Z is a parameter that indicates the normal functioning of the digestive system; η, γ, c_q take into account the amount of ballast substances and nutrients.

Next, we will study the effect of food and organic supplements. The food market today is very diverse. We consider some of the products that are necessary for the maintenance of life, presented in today’s food market. It was analyzed and found that foods contain a large number of chemicals that can adversely affect the body’s digestive system.

As for dietary supplements, the advertising of pharmaceutical companies plays a significant role here. None of the dietary supplements can give a full range of vitamins that give fresh fruit or vegetables. Only plant extracts are contained in dietary supplements. Supplements also have a narcotic effect on the body’s addiction. Or in the body, there is an effect of “impossible existence without this dietary supplement”.

Model (8) fully reflects the functioning of the digestive system without and with the use of various nutrients. Diagnosis can be performed on a dynamic model, using the concept of fractals. The body (in this case, the digestive system) is a functional digestive system B . The whole “factory” of the digestive system R . Thus, diagnosing the chemical metabolism of a particular system, we diagnose the whole body:

$$\log R = kd_n \forall \prod (\eta - \gamma) c_q \cdot \log \frac{1}{a_{(\eta-\gamma) \cdot c_q / b_j}} = Z \quad (8)$$

Thus, this model can be used in automated systems of diagnosis, solution. It is necessary to take into account the anthropometric indicators of the body: height, weight, as well as age and sex. Using heart rate monitor, on the one hand, it is possible to identify at earlier stages of the pathology of the esophagus; on the other hand, it is possible to determine the individuality of perception and intolerance of the body to various food products.

4.3 Experiment Reliability

In the work, the data validity is carried out. For this, the following statistical data analyzes were carried out the Student criterion calculation when testing the hypothesis of the mean equality to a given value of A, the Student criterion calculation when testing the hypothesis of the average equality with related samples, the Student criterion calculation when testing the hypothesis of the average equality without the assumption of variances, analysis of uniformity Kohren dispersions.

When choosing the criteria for checking the data reliability, a check was made on the normality of the random variables’ distribution using a macro built into the program Microsoft Excel 2016. The probability of data reliability was 95%.

These data processing criteria are selected considering the choice of solving specific problems, which will be described below, by the normal distribution and measurement scales of measurement sample values.

5 Experiment

5.1 Monitoring and Measuring

This study proposes monitoring by measuring heart rate on a wrist heart rate monitor. Since the tachycardia manifestation with GERD [10] is a frequent symptom, it can be assumed that for any esophagus irritation there should be at least slight changes in heart rate. We will use the Garmin Instinct and Garmin vivosport heart rate monitors; we’ll compare their performance with the Philips MX400 electrocardiograph.

As a result of measurements, it was found that certain patients’ during eating food that causes esophagus irritation, heart rate rises by 20–30 beats/minute, and while eating food that on probation doesn’t irritate the esophagus, heart



Fig. 2. The increasing of the heart rate during breakfast in a group with GERD

rate rises by 10 beats per minute; in the control group, heart rate changed within 5 units (Fig. 2, 3, 4, 5).

Symptoms were accompanied by sinus tachycardia, which, in turn, was practically not felt by the observed patients themselves. Individuals of the group had pain in the esophagus, and this symptom was noted, as a rule, when eating food with pH 3.3 and below, and the maximum peaks of heart rate were recorded when eating fatty foods.

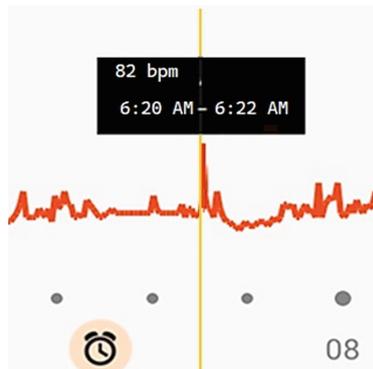


Fig. 3. WorldView-3 band combinations

At the same time, a high triglyceride value of 247 mg/dl was recorded. The control group had a temporary increase in heart rate, but it did not reach the threshold of 90 beats/min, so this is not tachycardia.

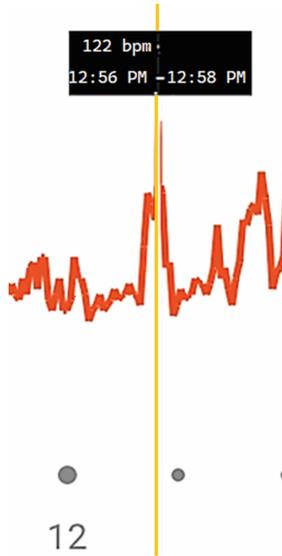


Fig. 4. The increasing of the heart rate during lunch for a group with GERD

Here again, the dependence on food intake was traced - the heart rate values were higher when taking fatty foods or taking a large amount of food. During the monitoring, individual perception of the group suffering from GERD on tea, cheese, bananas, and shrimps was revealed.

There was no perceptible difference in heart rate values during eating between lunch and dinner in the observed group. Additionally, an increase in heart rate is recorded before eating. This may indicate that the observed person took food at the wrong time and gastric juice was thrown into the esophagus (gastric reflux). Separately, the effect of antacid preparations after relapse of GERD was examined (Fig. 6).

Non-irritating food also leads to an increase in heart rate, possibly directly when passing through the damaged parts of the esophagus, but the values practically did not go beyond the value of 90 beats/min. It can also be said that the recovery of heart rate to normal in these patients was faster. After antacids use, a heart rate decrease occurs, the effect of the drug occurs after 2–5 min, the action duration takes about 30–40 min, then the heart rate rises slightly.



Fig. 5. The heart rate at the GERD group during dinner with non-irritating

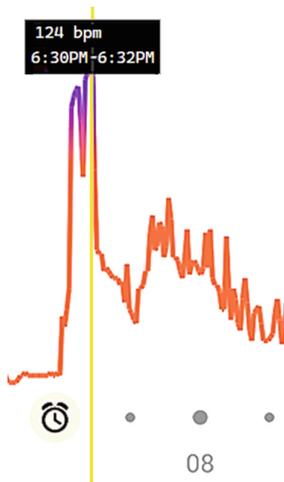


Fig. 6. Antacids using shows a heart rate decrease from 124 to 79

The use of antacids showed a rapid decrease in heart rate, as well as a long period of insensitivity of the heart rate to the food used. After a period of remission, there was an increase in heart rate by 20–30 units with a gradual decrease to normal. It was also noted that in some of the examined the acid is thrown into the esophagus after 23:00, as a result, there are jumps in heart rate during rest (Fig. 7, 8).

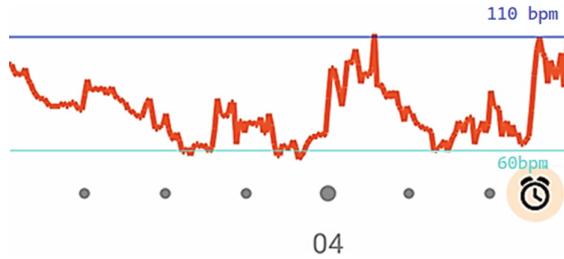


Fig. 7. GERD acid reflux

The majority of those observed in this group, during rest, feel a change in heart rate during the reflux of gastric acid into the esophagus. Moreover, the sensation of this symptom intensified with the position of the body on the stomach. Figure 2, 3, 4, 5, 6, 7 shows the average results for the entire group.

In some patients, the reverse dynamics were observed, after eating, heart rate decreased by about 50% within 2–4 min, which is possibly due to spasms of blood vessels, respectively, after eating warm food, the vessels expanded and the heart rate decreased. Figure 8 shows a 50% reduction in the heart rate during lunchtime.

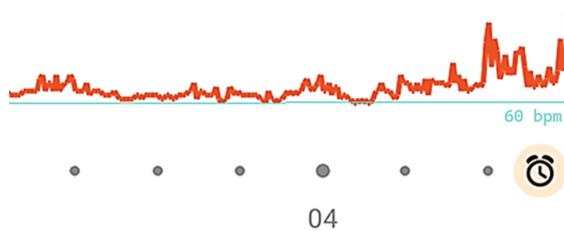


Fig. 8. The absence of sharp jumps in heart rate in a healthy group

In the control group, there was practically no significant change in the dynamics of heart rate at rest. In addition to moments of psychoemotional stress in some of the observed, which is recorded during sleep. It can also be noted that psychosomatic disorders are one of the main reasons for the development of gastrointestinal pathologies [25]. During an exacerbation of the disease (not necessarily associated with the gastrointestinal tract), the heart rate also increases by an average of 10–15 beats/min at rest.

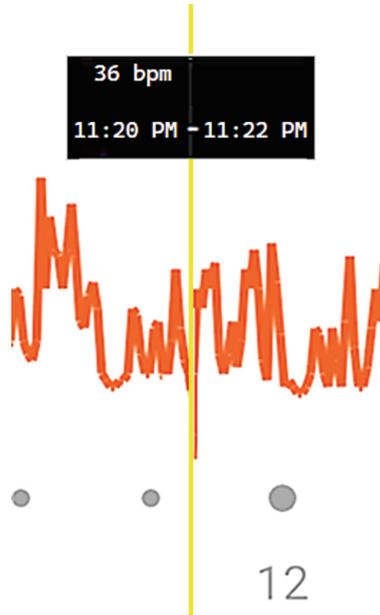


Fig. 9. The heart rate reduction during lunch meal

5.2 Binding Between Variables and Data Validity

The observed decrease in heart rate was recorded in 3 patients, in addition to the possible relief of vasospasm, it is possible to assume the registration of extrasystoles, but to confirm/refute this hypothesis, a full-fledged Holter monitoring is required.

During comparing the heart rate received by a photoplethysmogram method with a heart rate measured by a Philips MX400 electrocardiograph, we've obtained the following data: $\Delta\%$ of the studied heart rate monitors at average heart rates does not exceed 1.5%, at high heart rates $\Delta\%$ reaches 5%. Based on these data in the opinion of the authors, we can use a heart rate monitor for monitoring food perception during GERD.

There is also confidence that a universal diet for GERD sufferers does not exist, in each individual case, the trigger may be its own product, and only monitoring can identify this product. We analyzed the relationship between the variables and the reliability of the data. Two values are used - the monitoring data of the electrocardiogram and the heart rate monitor. The data is distributed according to a law different from the normal, both parameters are in the scale of relations. In this case, we will apply Spearman's nonparametric correlation criterion. Spearman's coefficient is calculated by the formula:

$$\rho(A, B) = 1 - \frac{6 \cdot \sum_{i=1}^n (R_{1i} - R_{2i})^2}{n^3 - n} \tag{9}$$

where R_{1i} and R_{2i} are the ranks of the i -th object for each of the compared variables. Let us present the monitored data below (Table 2).

Table 2. Analysis of the relationship between variables and data reliability

Nº	HR sensor Garmin Instinct	Philips MX400	Ranks R_{1i}	Ranks R_{2i}	$(R_{1i} - R_{2i})^2$
1	95	107	24	24	0
2	112	118	25	26	1
3	119	124	27	27	0
4	138	146	29	29	0
5	143	160	30	30	0
6	135	128	28	28	0
7	112	116	25	25	0
8	94	102	23	23	0
9	77	78	18	18	0
10	75	75	17	17	0
11	82	84	22	22	0
12	78	81	20	21	1
13	78	80	20	20	0
14	77	79	18	19	1
15	68	71	16	16	0
16	65	67	13	12	1
17	64	65	6	3	9
18	67	70	15	14	1
19	63	68	1	13	144
20	65	70	13	14	1
21	63	65	1	3	4
22	64	66	6	10	16
23	64	65	6	3	9
24	63	64	1	1	0
25	64	65	6	3	9
26	64	65	6	3	9
27	63	65	1	3	4
28	64	66	6	10	16
29	64	65	6	3	9
30	63	64	1	1	0
					235
					0.36

Substituting the obtained data from the table into the Spearman criterion formula, we obtain the value 0.95. According to the critical values of Spearman's rank correlation at a significance level of 0.05, we obtain a tabular value of 0.36. Given that the calculated value is greater than the tabular value, the relationship between the variables is significant and the reliability of the data is 95%. It should also be noted that a chest heart rate monitor will give more accurate results [26].

6 Discussion

The work was carried out through statistical data processing in order to find the reliability of the data. The following data analyzes were carried out: check data scatter within a sample, comparison of the indicators of the control and experimental samples, comparison of the indicators of the sample before and after the experiment, can we assume that the same indicator value takes place in several samples?

1. Data scattering within the samples is carried out using Student’s criterion (10). The law of distribution is close to Gauss law.

$$t = \frac{(\bar{X} - A)\sqrt{N}}{S^2} \tag{10}$$

where N is the sample size; S^2 is the empirical variance of the sample; A is the estimated value of the average value; \bar{X} is the average value.

The null hypothesis of the means equality is rejected if: $|t| > t_{(\nu, \alpha/2)}$ [26].

The number of freedom degrees for the t -test is $V = N - 1$.

$t_{(\nu, \alpha/2)}$ - table value of Student criterion at 0.05 significance level.

The results shown in the Table 3 indicate that the data within the group are distributed according to one law. Where 1 is estimated value t before breakfast; 2 is during breakfast; 3 is before lunch; 4 is during lunch; 5 is before dinner; 6 is during dinner; 7 is before using antacids; 8 is after using antacids; 9 is control group before experiment; 10 is control group during experiment.

Table 3. Data validation

Veracity - 95%, significance level - 0.05	Estimated value t									
	1	2	3	4	5	6	7	8	9	10
Table value - (2.14)	1.4	1.4	1.5	1.4	1.5	1.5	1.4	1.3	1.3	1.2
Conclusion	There is no dispersion									

The results shown in the table indicate that the data within the group are distributed according to one law.

2. Let compare the sample before and after the experiment in the experimental sample. Data of distribution are closed to Gauss law. To analyze the dispersion, we choose Student’s criterion for related samples (11):

$$t = \frac{\frac{\sum_{i=1}^N (x_i - y_i)}{N}}{\sqrt{\frac{\sum_{i=1}^N (x_i - y_i)^2 - \frac{(\sum_{i=1}^N (x_i - y_i))^2}{N}}{N(N-1)}}} \tag{11}$$

where x_i and y_i are the values of the connected series of observations; N is the sample size (each, since they are the same). The number of freedom degrees for

the t-test is $V = N - 1$. The hypothesis of equality of means is rejected under the following conditions: $|t| > t_{(\nu, \alpha/2)}$ [26]. In our case, the following results were obtained during patients breakfast, $t = 0$ with a table value of -1.83 . Since the calculated value is larger than the tabulated value with a significance level of 0.05, it can be said with a confidence of 95% that there is dispersion of data before and after meals. The remaining comparative estimates are presented in Table 4.

Table 4. Comparative analysis of the sample before and after the experiment

Veracity - 95%,	Estimated value t			
Significance level - 0.05	During breakfast	During lunch	During dinner	After using Antacids
Table value - (2.14)	2.5	2.3	2	1.9
Conclusion	There is no dispersion			

3. Let conduct a comparative analysis of the control and experimental indicators samples.

Data of distribution are closed to Gauss law. To analyze the dispersion, we choose the Student criterion without the assumption of variances (12).

$$t = \frac{(\bar{X}_1 - \bar{X}_2)\sqrt{N}}{\sqrt{\left(\sum_{i=1}^N (X_{1i} - X_{2i} - (\bar{X}_1 - \bar{X}_2))^2\right)/(N-1)}}. \quad (12)$$

where N is the size of the first and second samples; \bar{X}_1 and \bar{X}_2 estimate of average values; X_{1i} and X_{2i} are the current values of the variables respectively.

The number of freedom degrees for the t-test is $V = N - 1$. The hypothesis of equality of means is rejected under the following conditions: $|t| > t_{(\nu, \alpha/2)}$. The analysis data are presented in Table 5.

Table 5. Comparative analysis of the control and experimental indicators samples

Veracity - 95%,	Estimated value t		
Significance level - 0.05	Comparison of the experimental at breakfast and the control groups	Comparison of the experimental at lunch and the control groups	Comparison of the experimental at dinner and the control groups
Table value - (2.14)	58.24	57.05	58.12

4. Let verify the dispersion of the data in several samples. Data of distribution are closed to Gauss law. Given that the samples are of equal size, the Kohren's criterion is applicable (13). Kohren's criterion is used for the same sample size.

$$G = \frac{S_{max}^2}{\sum_{i=1}^k S_i^2} \tag{13}$$

where S_{max}^2 is the maximum of variances; S_i^2 are the empirical variances calculated in each sample.

The result is compared with the table. If $G_{calc} < G_{t,\alpha,k,n-1}$, then the homogeneity hypothesis is accepted. Here n is the number of experiments by which the estimate of the dispersion of S_i is calculated [26]. Comparison is carried out in experimental samples when receiving examined antacids and a control sample.

The estimated value of Kohen’s criterion was 0.34, while the tabular value was 0.32. The reliability of the data was 95%. Since the calculated value is larger than the tabulated one, with a reliable probability of 95% we can say that there is a dispersion of the results between the samples.

For more reliable monitoring of GERD with further diagnosis, machine learning of the biotechnics system was carried out. The biotechnics system includes two components: biological - a doctor and a patient, technical - heart rate monitor, personal computer. Using cluster analysis (Fig. 10), the system was trained to correctly classify images; in this case, it classified patients into “conditionally healthy” and “potentially sick”.

Regression analysis assessed the reliability of the constructed models using machine learning. According to the database we proposed, the program itself provided cluster and regression analysis methods.

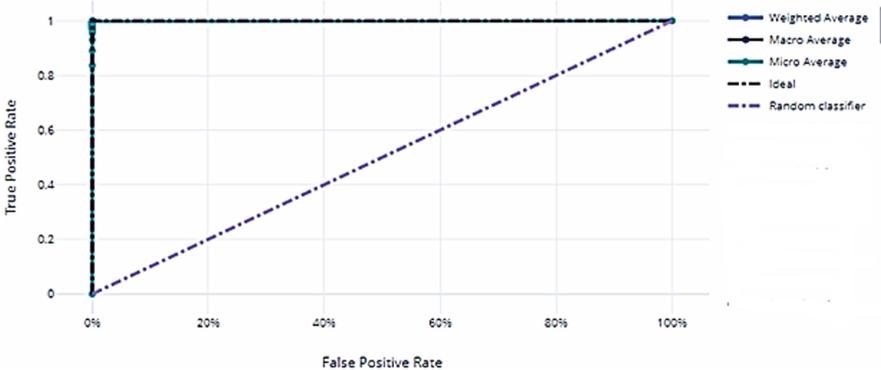


Fig. 10. One of the windows for classifying data in machine learning

The machine is properly trained and classifies the image. The biological part of the biotechnical system is represented by two groups - “conditionally healthy” and “potentially sick” patients. Regression analysis showed the reliability of the model built using machine learning.

After conducting experiments and subsequent analysis of the results, we obtained an accuracy of 99%. The system has correctly learned to classify data. A regression analysis showed an almost linear regression. Anomaly detection analysis is shown in Fig. 10, 11. First, we train the model.

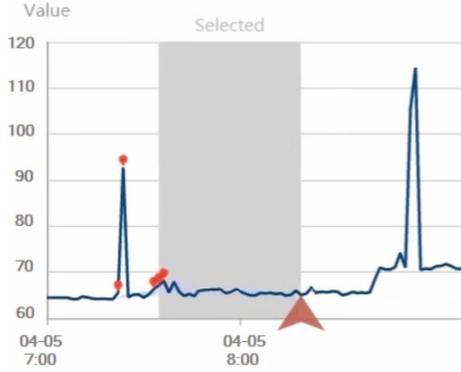


Fig. 11. Neural network learning

While learning the neural network, the system indicates the detected anomalies. As you can see, the system records the deviation from the isoline as an anomaly, however, a deviation of 5 units is not recorded as an anomaly. Next, we test our model. An anomaly detection component was used in Microsoft Azure Machine learning to analyze the accuracy of the research results.

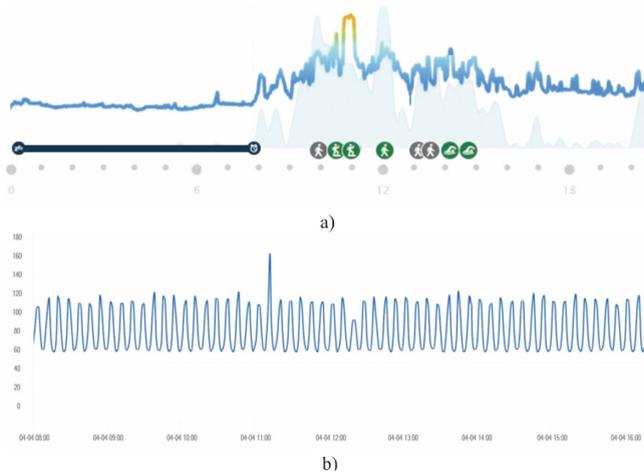


Fig. 12. a) Heart rate monitoring logging to Garmin connect; b) analyzed data using the anomaly detection component

Figure 12a) shows heart rate monitoring logging to Garmin connect. Figure 12b) shows the analyzed data using the anomaly detection component. As you can see, the anomaly detection model revealed two anomalies: the first anomaly at an amplitude of 162 bpm, the second anomaly at an amplitude of 90 bpm, which matches the Garmin connect data. Thus, experimental studies have shown the individual perception and reaction of the cardiovascular system of people in remission to food. The chosen method for assessing the reaction of the heart to a stimulus in the form of food and drugs is an easy tool for detecting gastrointestinal diseases in the early stages.

7 Conclusions

Gastroesophageal reflux is a condition in which gastric contents enter the esophagus when the cardiac opening is abnormally open. The main treatment for uncomplicated GERD consists of a healthy lifestyle and proper nutrition. In the presence of inflammatory processes in the esophagus in the form of reflux esophagitis, Barrett's esophagus, it is necessary to use medication, and in other cases, surgical treatment.

The study shows that during the period of remission in people with a diagnosis of "reflux" there is not only an individual perception of food but also a reaction from the cardiovascular system. This is observed in the course of experiments. By the reaction of the heart to an irritant (for example, food or drugs), one can judge the presence or absence of certain gastrointestinal diseases.

In the course of work, it was found that wrist pulsometers can be used to establish the susceptibility of a particular food product. For a more accurate analysis, software automation of the process is required.

It was also found that with a small to medium load on the heart, wrist heart rate monitors have little dispersion compared to a reference device.

Statistical data processing was carried out in order to find the information received reliability.

To select the criteria for determining the dispersion of the results, it was determined that the data of the samples are distributed according to the normal or close to normal distribution law. Processing data prove that a wrist heart rate monitor can be used to monitor GERD.

In our future work we plan to carry out similar research on the example of another disease to confirm the applicability of the techniques and technologies obtained in this research.

References

1. Acid reflux - american college of gastroenterology (2019). <https://gi.org/topics/acid-reflux/#tabs2>
2. Awesome satellite imagery datasets, github (2019). www.cs.tut.fi/samba/Publications

3. Microsoft Computer Generated Building Footprints for Canada (2019). <https://github.com/Microsoft/CanadianBuildingFootprints>
4. Alcantarilla, F. Solutions, T.: Fast explicit diffusion for accelerated features in non-linear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell.* **34**(7), 1281–1298 (2011). <https://doi.org/10.5244/C.27.13>
5. Audebert, N., Le Bertrand, S., Lefèvre, S.: Deep learning for remote sensing - An introductio. IRISA - Université Bretagne Sud, Atelier DLT Sageo, ONERA (2018)
6. Babichev, S., Korobchynskiy, M., et al.: Development of a technique for the reconstruction and validation of gene network models based on gene expression profiles. *East.-Eur. J. Enterprise Technol.* **1**(4–91), 19–32 (2018). <https://doi.org/10.15587/1729-4061.2018.123634>
7. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopoly. Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
8. Câmara, G., et al.: SPRING: integrating remote sensing and GIS by object-oriented data modelling. *Comput. Graph.* **20**(3), 395–403 (1996). [https://doi.org/10.1016/0097-8493\(96\)00008-8](https://doi.org/10.1016/0097-8493(96)00008-8)
9. Deng, J., et al.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
10. Frazier, K.: *The Acid Reflux Escape Plan*. Exisle Publishing (nz), Wollomb. A (2015)
11. Gawron, A.J., French, D.D., Pandolfino, J.E., Howden, C.W.: Economic evaluations of gastroesophageal reflux disease medical management. *Pharmacoeconomics* **32**(8), 745–758 (2014). <https://doi.org/10.1007/s40273-014-0164-8>
12. Han, J., Zhang, D., Cheng, G., Guo, L., Ren, J.: Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Trans. Geosci. Remote Sens.* **53**(6), 3325–3337 (2015). <https://doi.org/10.1109/TGRS.2014.2374218>
13. Hazirbas, C., et al.: FuseNet: incorporating depth into semantic segmentation via fusion-based CNN architecture. In: *Asian Conference on Computer Vision*. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-54181-5_14
14. Henderson, F.M., Lewis, A.J.: *Principles and applications of imaging radar. Manual of remote sensing: Third edition 2* (1998)
15. Higgins, P.: *No More Bile Reflux: How to Cure Your Bile Reflux and Bile Gastritis Naturally Without Medications*. Amazon Digital Services LLC (2017)
16. Hnatushenko, V.V., Kashtan, V.J., Shedlovska, Y.I.: Processing technology of multispectral remote sensing images. In: *IEEE International Young Scientists Forum on Applied Physics and Engineering YSF-2017*, pp. 355–358 (2017). <https://doi.org/10.1109/YSF.2017.8126647>
17. Hnatushenko, V.V., Vasyliiev, V.V.: Remote sensing image fusion using ICA and optimized wavelet transform, *International archives of the photogrammetry, remote sensing & spatial information sciences*. In: *XXIII ISPRS Congress, Prague, Czech Republic*, vol. XLI-B7, pp. 653–659 (2016). <https://doi.org/10.5194/isprs-archives-XLI-B7-653-2016>
18. Hordiiuk, D.M., Hnatushenko, V.V.: Neural network and local Laplace filter methods applied to very high resolution remote sensing imagery in urban damage detection. In: *IEEE International Young Scientists Forum on Applied Physics and Engineering (YSF)*, pp. 363–366 (2017). <https://doi.org/10.1109/YSF.2017.8126648>

19. Kaiming, H., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016). <https://doi.org/10.1109/CVPR.2016.90>
20. Kaiming, H., et al.: IEEE International Conference on Mask R-CNN, Computer Vision (ICCV). IEEE (2017). <https://doi.org/10.1109/ICCV.2017.322>
21. Krizhevsky, A., Sutskever, I., Geoffrey Hinton, E.: ImageNet classification with deep convolutional neural networks. *Adv. Neural. Inf. Process. Syst.* (2012). <https://doi.org/10.1145/3065386>
22. Lipka, S., Kumar, A., Richter, J.E.: No evidence for efficacy of radiofrequency ablation for treatment of gastroesophageal reflux disease: a systematic review and meta-analysis. *Clin. Gastroenterol. Hepatol.* **15**13(6), 1058–1067 (2014). <https://doi.org/10.1016/j.cgh.2014.10.013>. PMID2545955
23. Longbotham, N., et al.: Prelaunch assessment of worldview-3 information content. In: 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). IEEE (2014). <https://doi.org/10.1109/WHISPERS.2014.8077566>
24. Lu, X., Yuan, Y., Zheng, X.: Joint dictionary learning for multispectral change detection. In: *IEEE Transactions on Cybernetics*, vol. 47(4), pp. 884–897 (2017). <https://doi.org/10.1109/TCYB.2016.2531179>
25. Mayev, I.V., Kazulin, A.N., Yurenev, G.L.: Cardiac syndrome under gastroesophageal reflux disease. Manifestations, Incidence Rate, Causes and Remedies. Effective pharmacotherapy. *Cardiol. Angiol.* **3**(25) (2013)
26. Montgomery, E.: *Reproducibility in Biomedical Research*, 1st edn. Academic Press, Massachusetts (2020)
27. Samsudin, H.S., Shafri, H.Z., Hamedianfar, A.: Development of spectral indices for roofing material condition status detection using field spectroscopy and WorldView-3 data. *J. Appl. Remote Sens.* **10**(2) (2016). <https://doi.org/10.1117/1.JRS.10.025021>
28. Szabó, S., Gacsi, Z., Boglárka, B.: Specific features of NDVI, NDWI and MNDWI as reflected in land cover categories. *Acta Geographica Debrecina Landscape Environ.* **10**(3–4), 194–202 (2016). <https://doi.org/10.21120/LE/10/3-4/13>
29. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015). <https://doi.org/10.1109/CVPR.2015.7298594>
30. Testoni, P.A., Vailati, C.: Transoral incisionless fundoplication with esophyx for treatment of gastroesophageal reflux disease. *Dig. Liver Dis.* **12**44(8), 631–635 (2012). <https://doi.org/10.1016/j.dld.2012.03.019>. PMID22622203
31. Wassenberg, J.: *Efficient Algorithms for Large-Scale Image Analysis. Schriftenreihe automatische Sichtprüfung und Bildverarbeitung.* KIT Scientific, Publ (2011). <https://doi.org/10.5445/KSP/1000025309>
32. Xin, H., Zhang, L.: An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. In: *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51.1, pp. 257–272 (2013). <https://doi.org/10.1109/TGRS.2012.2202912>
33. Xiuping, J., Richards, A.: Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification. *IEEE Trans. Geosci. Remote Sens.* **37**.1, 538–542 (1999). <https://doi.org/10.1109/36.739109>
34. Zhang, J.: Multi-source remote sensing data fusion: status and trends. *Int. J. Image Data Fusion* **1**(1), 5–24 (2010). <https://doi.org/10.1080/19479830903561035>



Processing Technology of Thematic Identification and Classification of Objects in the Multispectral Remote Sensing Imagery

Volodymyr Hnatushenko , Yana Shedlovska ^(✉) , and Igor Shedlovsky 

Dnipro University of Technology, 19 Dmytra Yavornytskoho Ave.,
Dnipro 49005, Ukraine

{Hnatushenko.V.V,Shedlovska.Y.I,Shedlovskii.I.A}@nmu.one

Abstract. This paper is devoted to an automated processing technology for remote sensing data of high spatial resolution. The developed technology is based on an object-based approach, which allows the classification, analysis and identification of individual objects on the Earth's surface by taking into account their properties. The proposed processing technology includes the following key steps: pre-processing, segmentation, identification of different types of objects, and classification of the whole image. The multiscale segmentation method was used to obtain objects for analysis. The features of an image that allow one to accurately identify different types of objects were calculated: geometric, spectral, spatial, texture, and statistical features. On the basis of the calculated features, a decision on the object class is made. A model based on fuzzy inference is chosen to decide on the classes of image segments. The general accuracy, which shows the percentage of correctly classified pixels, and the Kappa index were used to evaluate the classification results.

Keywords: Remote sensing data · Segmentation · Classification · Image features · Kappa index · Satellite imagery · Image recognition

1 Introduction

The classification of different types of the Earth surface and the identification of particular objects is a challenging task. Earlier a lot of algorithms for image classification were proposed. Different algorithms were developed for processing of data received by satellites and aircraft. These spacecraft and aircraft are distinguished by the type of obtained data, physical means of information fixation, and the spectral and spatial characteristics of the data [21]. This work is devoted to such tasks of remote sensing imagery processing as thematic classification and identification of objects. The task of identification is to find individual objects or groups of objects of the same type in a satellite image [32].

The *aim of image classification* is to obtain a thematic map of the Earth's surface. Each of the image pixels is represented by a vector of values - x_{ij} in N

spectral bands. The problem of image classification is to define the class the pixel belongs to. Classes are determined in advance and correspond to different types of objects on the Earth's surface.

The tasks of classification and identification of objects are closely related because to obtain a thematic map of the area, it is necessary to accurately identify all types of objects present in the image [28]. A processing technology of aerial images and satellite images of high and ultrahigh spatial resolution is developed in this work, which allows one to classify the whole image or to identify objects of certain classes. The images used in this work obtained by the WorldView-2 and WorldView-3 satellites have a spatial resolution of 1.84 m and 1.24 m in multispectral bands and 46 cm and 31 cm in panchromatic bands, respectively [11]. Ultra-high spatial resolution allows us to accurately identify the boundaries of objects on the Earth's surface and take into account their geometric features.

2 Related Works

There are different methods of deciding on the object class. The choice of classification method depends on the problem and type of the data to be classified. The most widely used classification methods also could be applied to remote sensing data. To determine which classification method is the best for the current task is a topical problem [30].

The classical methods are divided into two groups: methods of controlled classification and methods of uncontrolled classification. For *controlled classification*, reference areas are used, which are selected by the operator in accordance with their belonging to a certain class [10]. For each reference region, spectral features that represent one class of pixels in the image are determined. Each pixel is compared with all the reference features that belong to different classes. In controlled classification, information classes and their number are determined first, and then the corresponding spectral classes. The most commonly used algorithms of controlled classification are: the algorithm based on determining the smallest distance, the parallelepiped algorithm, and the maximum probability algorithm [4].

In the case of *uncontrolled classification*, spectral classes are first distinguished on the basis of the information presented in the data, and only then they are compared with the known distribution of real classes. Data are grouped using cluster analysis algorithms. The initial data of the algorithm indicate the number of clusters, cluster resolution criteria and possible variance of values in the cluster. The clustering process is iterative.

Algorithm clustering K-means refers each pixel of the image to one of the predefined clusters. Random pixel images are selected as the initial centers of the clusters. Each pixel of the image belongs to the cluster the distance to which is smallest. The algorithm minimizes the standard deviation of pixels from the centers of clusters. At each iteration, new cluster centers are calculated and image pixels are distributed between the clusters in a new way. The process is

repeated until the clusters remain unchanged. The ISODATA algorithm (Iterative self-Organizing Data Analysis Technique) is a modified K-means algorithm. It consists in merging clusters if their resolution is below a threshold and in splitting a cluster if it becomes too large.

In [3] the remote sensing image classification was performed by combining support vector machine (SVM) and k-nearest neighbor (KNN). This was based on the separability of classes using SVM and the spatial and spectral characteristics of remote sensing data. A distance formula was proposed as the measure criterion that considers both luminance and direction of the vectors.

Neural networks are widely used in solving classification problems [25]. The vectors of pixel values in different spectral ranges of the image are used as the input data, as a result of the network operation the pixel belongs to one of the classes. Network training is based on pixels of reference samples. There are many neural network architectures, which differ in the number of layers, the number of neurons, the number of connections, the type of transfer function [12]. The choice of neural network architecture depends on the problems to be solved [19]. However, the problem of this method is the need for a large number of training samples.

The decision tree classifier can be used for remote sensing image classification; it performs multistage classifications by using a series of binary decisions to place pixels into classes. Each decision divides the pixels in a set of images into two classes based on an expression. The data from many different sources and files can be used to make a single decision tree classifier. Input data can be from various sources and data types.

In [14] a fuzzy inference system was used to classify satellite images. Fuzzy output systems were used to classify satellite images obtained by satellites: GeoEye-1, QuickBird [14], Landsat-8 [36], ICONOS [20]. Fuzzy inference is an approximation of the input-output relationship based on a fuzzy knowledge base and operations on fuzzy sets [9].

3 Proposed Technology of Thematic Classification and Identification of Objects in the Remote Sensing Imagery

The proposed technology is based on the object-based approach to image analysis [23]. Today, the object-based approach is one of the most effective means of high spatial resolution imagery classification and analysis [34]. In contrast to the pixel-based approach [26], the object-based approach does not process individual pixels: it processes groups of pixels, the so-called objects or segments, which are obtained from image segmentation. The object-based approach uses for classification not only vectors of individual pixel values, but also the properties of objects. Statistical, spatial, geometric, spectral and textural properties can be used [32]. The proposed technology consists of several stages of image processing and allows one to identify particular types of objects or classify the whole image.

Remote sensing imagery typically includes several common classes of objects and land cover, such as vegetation, water, soil, buildings, shadows, and roads. These objects can be semantically described based on their physical characteristics. The object-based approach allows us to take into account the properties inherent in different types of objects and surfaces. The presence of multispectral bands makes it possible to use spectral information about objects in images. The proposed processing technology of multispectral remote sensing imagery includes the following key steps: pre-processing, segmentation, identification of different types of the objects, and classification of the whole image.

In the proposed technology, at the stage of segmentation different methods can be used. The following segmentation methods were investigated in our previous work [32]: the mean-shift method, multiscale segmentation and segmentation based on the K-means clustering method. The quality of the above segmentation methods is evidenced by the works in which the methods used were also applied to the analysis of satellite and aerial images. There is no single objective criterion for segmentation quality assessment for segmentation methods. The choice of the best method depends on the tasks. Also, to compare the quality of segmentation methods, there are image databases for which “reference” segmentation is known [40]. In our previous work we found that the multiscale segmentation method is most suitable for high spatial resolution remote sensing imagery processing [32].

Before applying the classification method, the features of image objects must be calculated. The features used in our work can be divided into the following categories:

- Geometric features;
- Spatial features;
- Spectral features;
- Statistic features;
- Textural features.

On the basis of calculated features the decision about the class of the object is made. Different classification methods could be applied for assigning classes to image objects. The choice of classification methods depends on the parameters calculated from the image and their characteristics. The decision about class assigning could be made on the basis of logic rule set or fuzzy inference system. The supervised classification methods could be applied to the object feature vectors to assign class to them. A model based on fuzzy inference is chosen to decide on the classes of image segments. In fuzzy inference systems, a decision on the object class is made on the basis of a set of fuzzy rules and fuzzy logic [14]. The rules are formulated taking into account the physical properties of each class. The property vectors of the segments are taken as the input information for the fuzzy output system. To do this, the properties of the image objects obtained at the segmentation stage were calculated. The structure of the proposed technology is presented in Fig. 1.

3.1 Pre-processing of Remote Sensing Data

Processing of remote sensing data can be divided into two important stages: pre-processing and thematic processing. Radiometric and geometric distortions

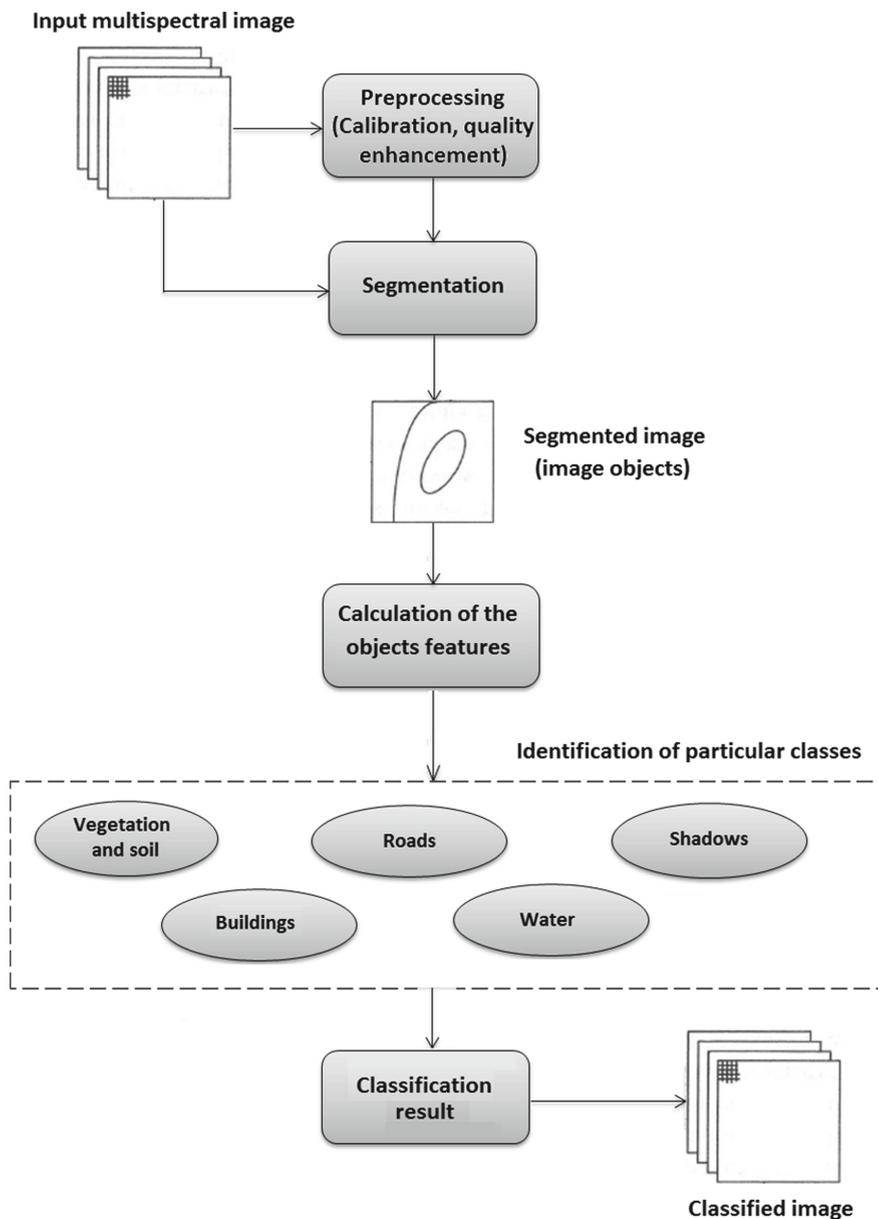


Fig. 1. The structure of the proposed technology

are eliminated from the data during pre-processing [17]. Enhancement of the image quality makes it more convenient for analysis and processing by automatic methods. At the stage of thematic processing, the information requested by the user is extracted for analysis.

There are the following types of pre-processing:

- geometric correction of images;
- radiometric calibration;
- correction of atmospheric influence;
- recovery of missed pixels;
- elimination of noise in images;
- increasing of spatial resolution.

Pre-processing methods also include shadow compensation. For some tasks, the presence of shadows in the image is undesirable [33]. Shadow compensation technology consists of several stages. One of the stages is the identification of shadows [7]. The existing methods of shadow identification were investigated in order to find out which of the methods gives the best result for the processed images.

At the stage of pre-processing, radiometric correction and increase of spatial resolution of input data were performed. To increase the spatial resolution, 8 multispectral image bands were merged with the panchromatic band. For this purpose, the technology developed in [16] was used.

3.2 Segmentation of Remote Sensing Imagery

Image segmentation is the process of splitting an image into homogeneous sections that contain pixels with similar characteristics. Image segmentation is a key stage of image processing, which affects the further calculation of the image object properties and the result of classification [37]. To get the most accurate result, it is necessary to choose the best segmentation method and its parameters for the problem to be solved [24].

Segmentation methods can be divided into automatic and interactive, requiring user participation. Automatic methods are also divided into two classes: 1) selection of image areas with certain properties specific to a particular subject area (marker methods, binarization); 2) splitting the image into homogeneous areas. Methods that divide images into homogeneous areas are the most universal, as they are not focused on one subject area and specific analysis tasks. Such algorithms are the most common in the field of computer vision, they include methods of watershed, the gradient method, methods based on clustering of a multidimensional histogram [24].

In the proposed technology, at the stage of segmentation *multiscale segmentation method* was used. Multiscale segmentation method is based on the technique of sequential merging of adjacent image elements. This is an optimization procedure that minimizes the average heterogeneity of image objects [5].

Segmentation begins with a single image object and sequentially combines it with neighboring objects until the upper threshold of homogeneity is reached.

When the threshold is reached and the object can no longer be combined with neighboring objects, one of the neighboring objects becomes the initial and the procedure is repeated. The process is repeated cyclically until the merging of image objects is possible. The criterion of homogeneity is defined as a combination of spectral homogeneity and form homogeneity. In a given d -dimensional feature space, two image objects f_{1d} and f_{2d} are considered similar if their features in the feature space are close. The degree of similarity of objects can be expressed as:

$$h = \sqrt{\sum_d (f_{1d} - f_{2d})^2}. \quad (1)$$

In the case of combining image objects into one larger object, the average spectral homogeneity of the whole image decreases. The aim of the optimization procedure is to minimize the reduction of homogeneity associated with the processing of each image object. The object merges with the neighboring object if their combination results in the minimal reduction of homogeneity.

The degree of compatibility of two adjacent objects with the homogeneity h_1 and h_2 is determined by the change in the homogeneity h_{diff} before and after the potential union h_m . After merging h_{diff} should be minimal.

$$h_{diff} = h_m - \frac{h_1 + h_2}{2}. \quad (2)$$

When the value of h_{diff} exceeds a specified threshold, the merging of the objects does not occur. The threshold of merging is the main parameter of segmentation, it is usually set by the user. This parameter is also called a *scale parameter* because it affects the size of the segments. The higher the scale parameter; the larger the segments, the lower the scale parameter, the smaller the segments. This parameter is a combination of form homogeneity h_f and color homogeneity h_c .

The following types of homogeneity are taken into account in the segmentation process:

- 1) color homogeneity: the sum of the average pixel value deviations of objects in all spectral bands;
- 2) form homogeneity: the relative deviation of the length of the boundary of the object relative to the length of the most compact figure; it is determined by the following criteria:
 - a) smoothness criterion: the ratio of the length of the boundary of the object l to the length of the boundary of the bounding box rectangle b :

$$S = \frac{l}{b}. \quad (3)$$

- b) compactness criterion: deviation from the ideal shape, determined by the ratio of the length of the object l to the square root of the size of the object n :

$$C = \frac{l}{\sqrt{n}}. \tag{4}$$

The form homogeneity h_f depends on the criteria of smoothness S (smoothness) and compactness C (compactness) as follows:

$$h_f = (1 - w_f) S + w_f C. \tag{5}$$

The criterion of homogeneity h for each image object is determined by the form homogeneity h_f and color homogeneity h_c :

$$h = (1 - w_c) h_c + w_c h_f, \tag{6}$$

where w_c and w_f are the weights. The method of multiscale segmentation was applied to a test fragment of a satellite image (Fig. 2(a)). In the Fig. 2(b) the result of image segmentation is shown.

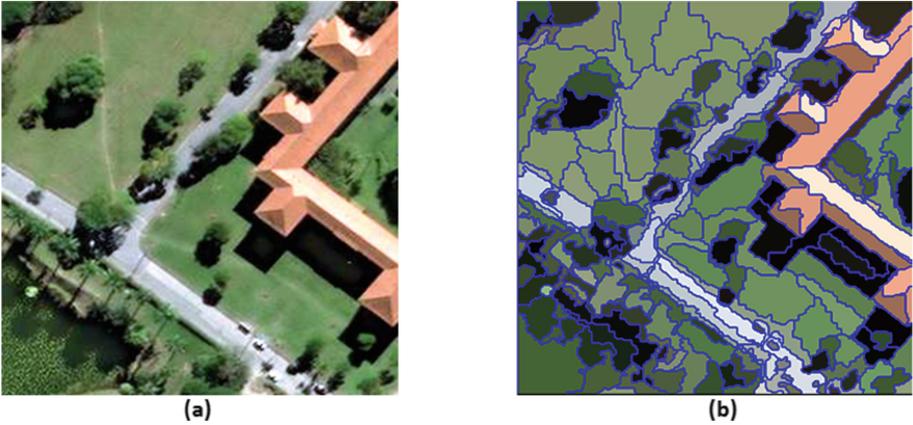


Fig. 2. (a) the original image; (b) the result of image segmentation

3.3 Classification

Classification is the procedure for making decisions on whether an object belongs to one of the classes. Different classification methods could be applied for assigning classes to image objects. The choice of classification methods depends on the parameters calculated from the image and their characteristics. A decision on the class of an object is made on the basis of certain features of the object. A set of features was used for each class recognition. A model based on fuzzy inference is chosen to decide on the classes of image segments [7]. In fuzzy inference systems, a decision on the object class is made on the basis of a set of fuzzy rules and fuzzy logic [14].

The rules are formulated taking into account the physical properties of each class. In practice, image objects have properties that are not unique to their class. The fuzzy inference system allows the classification of image objects when a decision on the class is impossible only based on the threshold value of one of the properties. Fuzzy logic operations used in fuzzy inference systems allow one to replace the crisp values of the properties of image objects with linguistic variables described by fuzzy membership functions. A linguistic variable is a variable that takes on a value from a set of words or phrases of a natural or artificial language [9]. The set of possible values of a variable is called a term set. In traditional classification methods, each pixel or segment is assigned to one of the classes; its belonging to a class can be described by an attribute that takes the value 0 or 1. In fuzzy inference systems, the belonging of a segment to a class is determined by the membership function. The membership function takes values from the range [0, 1]. If the function takes the value 1, the segment with the highest probability belongs to the class and does not belong to it if the function takes the value 0.

Identification of Vegetation and Soil. Objects belonging to the vegetation and soil classes are most easily identified by their spectral properties [8]. For the vegetation and soil classes identification the spectral properties of the segments were used. The values of the spectral indices were calculated, which allows one to accurately identify vegetation [39].

NDVI (Normalized Difference Vegetation Index) - normalized relative index of vegetation identification, it reflects the amount of plant biomass capable of photosynthesis [27]. It is one of the most common and accurate vegetation indices used in the processing of remote sensing data.

$$NDVI = \frac{NIR - RED}{NIR + RED}, \quad (7)$$

where *NIR* is the reflection of light in the near-infrared region of the spectrum, and *RED* is the reflection in the red region of the spectrum, respectively (bands 8 and 5 of images obtained by WorldView-2 and WorldView-3). The *NDVI* calculation is based on the two most stable regions of the spectral curve of the reflectivity of green plants. In the red part of the spectrum (0.6–0.7 μm) there is a maximum of absorption of solar radiation by chlorophyll of green plants, and in the infrared (0.7–1.0 μm) there is a maximum of radiation reflection. The index takes values from –1 to 1. For vegetation *NDVI* takes positive values: the more plant biomass, the higher it. The value of the index is also affected by the type of vegetation, its density, exposure, angle of inclination, and soil type under sparse vegetation. Dense and sparse vegetation corresponds to *NDVI* values from 0.7 to 1 and from 0.5 to 0.7, respectively. The standard deviation of the pixels of the object is calculated for all pixels belonging to the *k*-th segment of the image. The standard deviation is calculated separately for each image channel.

$$\sigma(k) = \sqrt{\frac{1}{N_k} \sum_{i=1}^{N_k} x_i^2 - \mu^2(k)}, \quad (8)$$

where N_k is the number of pixels in the k -th segment, x_i is the value of the i -th pixel of the segment. The standard deviation is used to define the texture nonuniformity of each segment. To identify grass, trees, and soil in images, we applied thresholding over the $NDVI$ histogram. Also the standard deviation was used to distinguish trees and grass classes. In the Fig. 3 the result of $NDVI$ index calculation is shown. The segments which belong to trees class are marked by the red lines.

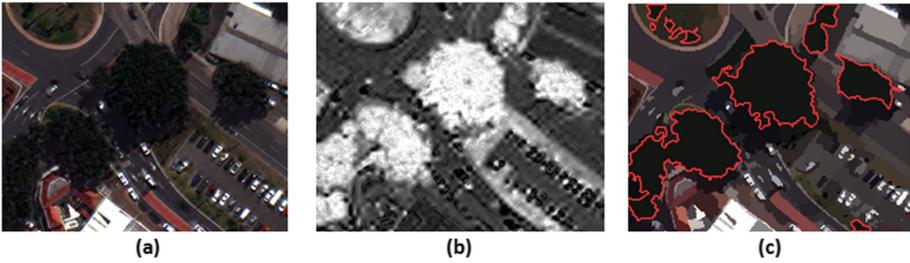


Fig. 3. (a) the initial image; (b) the result of the $NDVI$ calculation; (c) the result of the vegetation identification

Identification of Shadows. For shadow identification [31] the normalized saturation-value difference index ($NSVDI$) is used:

$$NSVDI = \frac{S - V}{S + V}, \tag{9}$$

where S is the saturation of the image, V is the brightness. To obtain the S and V image components, the image is transformed from the RGB color model to the HSV color model. $NSVDI$ takes values from -1 to 1 ; shadow areas have high index values. In the Fig. 4(b) the image of $NSVDI$ index is shown [35]. To identify shadow in images, we applied Otsu’s thresholding over the $NSVDI$ histogram. Then all pixels of the image are divided into shadow ones ($NSVDI > T$) and non-shadow ones ($NSVDI < T$). In the Fig. 4 the result of shadow identification is shown.

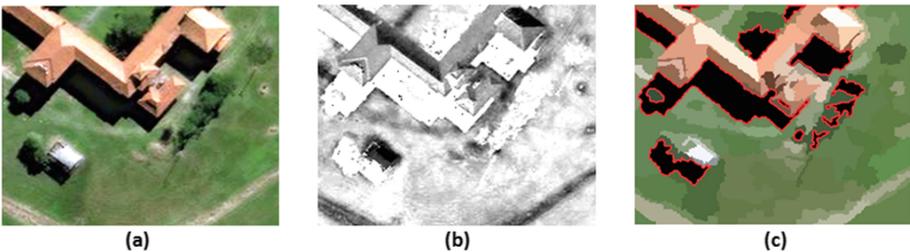


Fig. 4. (a) the initial image; (b) the result of the $NSVDI$ calculation; (c) the result of the shadow identification

Identification of Water. For water identification $NDWI$ is used. $NDWI$ (Normalized Difference Water Index) is a spectral index for the identification of water surfaces [29].

$$NDWI = \frac{GREEN - NIR}{GREEN + NIR} \quad (10)$$

where NIR is the reflection of light in the near-infrared part of the spectrum, and $GREEN$ is the reflection in the green part of the spectrum, respectively (bands 8 and 3 for images taken by WorldView-2 and WorldView-3). These spectral ranges were chosen to maximize the reflection of water surfaces of the green part of the spectrum and to minimize the low reflection of the waves in the near-infrared part of the spectrum. The index takes values from -1 to 1 ; water surfaces are characterized by positive $NDWI$ values [2]. In the Fig. 5 the result of water identification is shown.

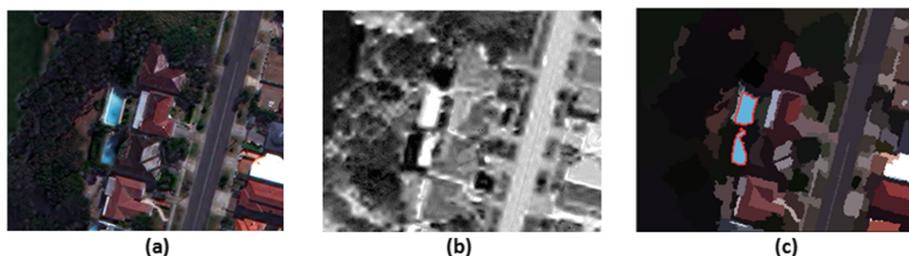


Fig. 5. (a) the original image; (b) the result of the NWDI calculation; (c) the result of the water identification

Identification of Buildings. A model based on fuzzy inference is chosen to identify image objects which belong to building class. The rules are formulated taking into account the physical properties of class. The feature vectors of the segments are taken as the input information for the fuzzy output system. The following features were used for building identification: the average value of the pixels, the object size, the rectangularity of an object, and the common boundary length.

The average value of the pixels of the object is calculated for all pixels belonging to the k -th segment of the image. It is calculated separately for each image band.

$$\mu(k) = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i, \quad (11)$$

where N_k is the number of pixels in the k -th segment, x_i is the value of the i -th pixel of the segment.

The object size is the number of pixels that make up the object. The area of the object can be calculated if the spatial resolution of the image by the number

of pixels is known. If information about the image is missing, the pixel area is taken as 1, the area of the object is equal to the number of pixels.

$$A_k = P_k \cdot u^2. \quad (12)$$

A_k is the area of the object; P_k is the number of pixels; k is the number of image segments that form the object; u^2 is the pixel size in units of distance in the image.

The rectangularity of an object is a property that shows how much the shape of the object corresponds to a rectangle. In order to calculate the degree of rectangularity of an object, it is necessary to construct a minimum bounding box for it. The software implementation of the construction of the bounding box is based on the algorithm presented in [13], which is widely used in the analysis of the shape of objects [15]. In this algorithm, a convex polygon of lines is built around the figure, one of the sides of which will be collinear to the side of the minimum bounding box.

The degree of object rectangularity R is estimated as the ratio of the area of the object A_k to the area of the bounding box A_R :

$$R = \frac{A_k}{A_R}, R \in (0, 1]. \quad (13)$$

Spatial properties describe the location of image objects and the presence of a common boundary between segments. To calculate the spatial properties, a matrix of adjacency of M image segments was constructed. The size of the matrix is k by k , where k is the number of image segments. Matrix element M_{ij} contains the length of the common boundary between image objects i and j .

The presence of a common boundary is a property that indicates whether the segments border each other.

The common boundary length is a property that shows the length of a common boundary between two adjacent segments. In the Fig. 6 the initial image and the result of building identification is shown.

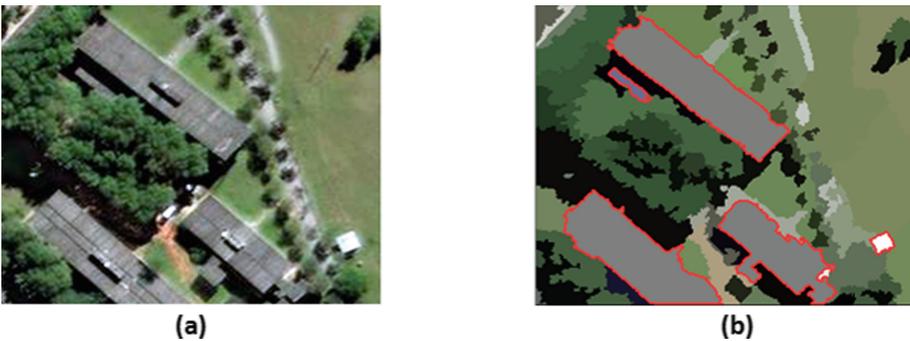


Fig. 6. (a) the initial image; (b) the result of the building identification

Identification of Roads. The geometric properties used for road identification describe the shape of the object (segment); they are calculated by pixels belonging to the object. A model based on fuzzy inference is chosen to identify image objects which belong to road class. The following features were used for road identification: the object compactness, the average value of the pixels, the ratio of the length to area, the boundary length, and the common boundary length.

The boundary length is the number of pixels lying on the object boundary. The boundary of an object characterizes its shape. The length of the object is the length of the larger side of the bounding box. Object width: The width of the object is the length of the smaller side of the bounding box.

The ratio of the length of the boundary to the area of the object is the length of the boundary of the object in units of distance in the image divided by A_k .

The object compactness is a numerical value that characterizes the degree of compactness of the figure [18]. The compactness of image objects is widely used for remote sensing data analysis [38]. This paper uses the well-known formula to determine the compactness:

$$C_{IPQ} = \frac{4\pi A_k}{B_k^2}, \quad (14)$$

where B_k is the length of the boundary of the object in units of distance in the image. Value range: $C_{IPQ} \in (0, 1]$. The larger the value of C_{IPQ} , the more compact the figure.

In the Fig. 7 the initial image and the result of road identification is shown.



Fig. 7. (a) the initial image; (b) the result of the road identification

Classification Result. As a result of image classification, the objects obtained after segmentation are assigned to one of the following classes: buildings, roads, trees, grass, soil, water, and shadows. Objects belonging to the vegetation and water classes are most easily identified by their spectral properties. A more difficult task is to identify roads and buildings because they have similar spectral properties of materials they are made of. This problem is especially topical for

densely built-up areas. Geometric properties and contextual information about the location of objects of other classes are used to identify buildings and roads. The classification results are combined to take into account the presence of shadows and vegetation in the neighborhood for the identification of houses and roads. In the Fig. 8 the initial image and the result of image classification is shown.

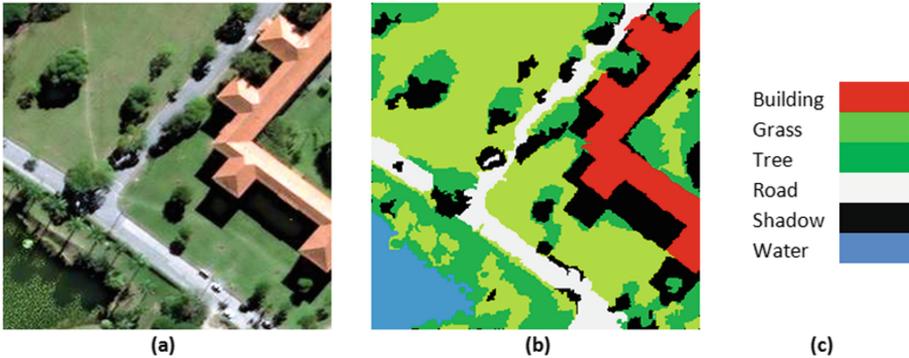


Fig. 8. (a) the original image; (b) the result of the classification; (c) color map

4 Classification Assessment

It is necessary to assess the accuracy of the classification, to determine how accurately the classes of objects in the image correspond to the real classes of objects on the Earth's surface. To do this, the classified image is compared with a reference class map, which is most often obtained by manual marking [1].

Statistical indicators were used to assess the accuracy of the classification [6]. According to the results of the classification, a *matrix of errors is formed*: a table that shows the classification error for each class [22]. The error matrix consists of n rows and n columns, where n is the number of object classes in the reference image. The rows of the matrix correspond to the classes on the reference map, and the columns correspond to the classes in the processed image. Denote by n_{ij} the pixels belonging to class j , but classified by the developed technology as class i . $n_{i(row)} = \sum_j n_{ij}$ is the number of pixels classified as class i by the developed technology, $n_{j(col)} = \sum_i n_{ij}$ is the number of pixels belonging to class j according to the reference map. The diagonal elements of the matrix are the pixels that are correctly classified.

The general accuracy O_c , which shows the percentage of correctly classified pixels, and the Kappa index - K were used to evaluate the classification results. The Kappa consistency index for two variants of the classification of N pixels

to k mutually exclusive classes was proposed by Cohen in [6].

$$O^c = \frac{1}{N} \sum_{i=1}^k n_{ii}, \tag{15}$$

where N is the number of pixels, k is the number of classes.

$$K = \frac{O^c - p_e}{1 - p_e}, \tag{16}$$

where

$$p_e = \frac{1}{N^2} \sum_i n_{i(col)} n_{i(row)}. \tag{17}$$

The overall accuracy O_c and the Kappa index take values from 1 to 0. The higher their value, the better the classification result. When using the overall accuracy, the evaluation result is affected by how often a particular class is found in a test sample. This is relevant when analyzing satellite images, in which one class can occupy a large area. For example, if class A occupies 90% of the image area and class B occupies 10%, then a randomly selected set of pixels with 90% probability will belong to class A. Then the total probability that these pixels will be correctly assigned to class A is randomly equal to 0, 81. Similarly, the total probability of correctly assigning pixels to class B at random is 0.01. As a result, even if you randomly assign pixels to a reference image, the overall accuracy of the map will be about 82%. That is, maps with several classes occupying large areas may have greater overall accuracy than maps with many classes of approximately the same area. The Kappa index is more indicative in such cases.

Table 1. Matrix of errors

Class	Building	Grass	Tree	Road	Shadow	Water
Building	8388	8	8	2	122	1
Grass	101	22237	1917	217	128	34
Tree	111	454	11339	154	1773	0
Road	175	325	57	8656	77	0
Shadow	30	11	963	2	14248	53
Water	46	0	0	0	4	5365

The Kappa statistics and the overall accuracy were calculated for the classified image (Fig. 8(b)). The results of the errors matrix calculation are shown in Table 1. For current image the overall accuracy $O_c = 0.91$ and the Kappa index = 0.88. For particular samples the classification accuracy can reaches higher values.

5 Conclusions

A new technology for decoding multidimensional satellite images based on the object-based approach has been developed, which allows the classification, analysis and identification of individual objects on the Earth's surface by calculating their properties. The presence of multispectral bands makes it possible to use spectral information about objects in images. Geometric, spatial, spectral, textural and static properties for identification of different types of objects on the Earth's surface are proposed. The proposed processing technology of multispectral remote sensing imagery includes the following key steps: pre-processing, segmentation, identification of different types of the objects, and classification of the whole image.

Depending on the type and quality of input data, different methods could be applied at the pre-processing, segmentation, and classification step. The multi-scale segmentation method was used to obtain objects for analysis. It is based on the technique of sequential merging of adjacent image elements with minimization of the average heterogeneity of image objects.

Different classification methods could be used in assigning classes to image objects. The choice of classification methods depends on the parameters calculated from the image and their characteristics. A classification model based on fuzzy inference was used to make a decision on classes of image segments. The rules are formulated taking into account the physical properties of each class. The property vectors of the segments are taken as the input information for the classification system. The set of the classes that were identified by the developed technology is as follows: buildings, roads, trees, grass, shadow, water.

Statistical indicators the overall accuracy O_c and the Kappa index were used to assess the accuracy of the classification. The results of statistics calculation showed the efficiency of the proposed technology.

References

1. Abdu, H.A.: Classification accuracy and trend assessments of land cover-land use changes from principal components of land satellite images. *Int. J. Remote Sens.* **40**(4), 1275–1300 (2019). <https://doi.org/10.1080/01431161.2018.1524587>
2. Acharya, T.D., Subedi, A., Yang, I.T., Lee, D.H.: Combining water indices for water and background threshold in Landsat image. In: *Proceedings*, pp. 143–148 (2018). <https://doi.org/10.3390/ecsa-4-04902>
3. Alimjan, G., Sun, T., Liang, Y., Jumahun, H., Guan, Y.: A new technique for remote sensing image classification based on combinatorial algorithm of SVM and KNN. *Int. J. Patt. Recogn. Artif. Intell.* **32**(7), 1859012 (23 pages) (2018). <https://doi.org/10.1142/S0218001418590127>
4. Amitrano, D., Guida, R., Ruello, G.: Multitemporal SAR RGB processing for Sentinel-1 GRD products: methodology and applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **5**(12), 1497–1507 (2019)
5. Chen, Y., Chen, Q., Jing, C.: Multi-resolution segmentation parameters optimization and evaluation for VHR remote sensing image based on mean nsqi and discrepancy measure. *J. Spat. Sci.* (2019). <https://doi.org/10.1080/14498596.2019.1615011>

6. Cohen, J.: Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychol. Bull.* **70**, 426–443 (1968). <https://doi.org/10.1037/h0026256>
7. De Souza Freitas, V.L., Da Fonseca Reis, B.M., Tommaselli, A.M.G.: Automatic shadow detection in aerial and terrestrial images. *Boletim de Ciências Geodésicas* **23**(4), 578–590 (2017). <https://doi.org/10.1590/s1982-21702017000400038>
8. Felegari, S., Sharifi, A., Moravej, K., Golchin, A., Tariq, A.: Investigation of the relationship between ndvi index, soil moisture, and precipitation data using satellite images. *Sustain. Agric. Syst. Technol.* 314–325 (2022). <https://doi.org/10.1002/9781119808565.ch15>
9. Gavrylenko, S.Y., Melnyk, M.S., Chelak, V.V.: Development of a heuristic antivirus scanner based on the file’s PE-structure analysis. *Inf. Technol. Comput. Eng.* **3**, 23–29 (2017)
10. Keshtkar, H., Voigt, W., Alizadeh, E.: Land-cover classification and analysis of change using machine-learning classifiers and multi-temporal remote sensing imagery. *Arab. J. Geosci.* **10**(6), 1–15 (2017). <https://doi.org/10.1007/s12517-017-2899-y>
11. Hnatushenko, V., Hnatushenko, V., Mozgovyi, D., Vasyliiev, V., Kavats, O.: Satellite monitoring of consequences of illegal extraction of amber in Ukraine. *Scientific bulletin of National Mining University. - State Higher Educational Institution “National Mining University”, Dnipropetrovsk* **158**(2), 99–105 (2017)
12. Hordiiuk, D.M., Hnatushenko, V.V.: Neural network and local Laplace filter methods applied to very high resolution remote sensing imagery in urban damage detection. In: *Proceedings of 2017 IEEE International Young Scientists Forum on Applied Physics and Engineering* (2017). <https://doi.org/10.1109/ysf.2017.8126648>
13. Huang, Z., Wang, F., You, H., Hu, Y.: Imaging parameters-considered slender target detection in optical satellite images. *Remote Sens.* **14**(6) (2022). <https://doi.org/10.3390/rs14061385>
14. Jabari, S., Zhang, Y.: Very high resolution satellite image classification using fuzzy rule-based systems. *Algorithms* **6**, 762–781 (2013). <https://doi.org/10.3390/a6040762>
15. Kampker, A., Sefati, M., Rachman, A.A., Kreisköther, K., Campoy, P.: Towards multi-object detection and tracking in urban scenario under uncertainties. In: *Proceedings of 4th International Conference on Vehicle Technology and Intelligent Transport Systems*, pp. 156–167 (2018). <https://doi.org/10.5220/0006706101560167>
16. Kashtan, V.J., Hnatushenko, V.V., Shedlovska, Y.I.: Processing technology of multispectral remote sensing images. In: *Proceedings of 2017 IEEE International Young Scientists Forum on Applied Physics and Engineering*, pp. 355–358 (2017). <https://doi.org/10.1109/ysf.2017.8126647>
17. Kidwai, F.Z., Ahmad, F.: Satellite image enhancement: a review. *Int. J. Res. App. Sci. Eng. Technol.* **7**(6), 573–576 (2019). <https://doi.org/10.22214/ijraset.2019.6100>
18. Li, W., Church, R., Goodchild, M.F.: The p-compact-regions problem. *Geogr. Anal.* **46**(3), 250–273 (2014). <https://doi.org/10.1111/gean.12038>
19. Lima, R.P.D., Marfurt, K.: Convolutional neural network for remote-sensing scene classification: transfer learning analysis. *Remote Sens.* **1**(12), 86 (20 pages) (2020). <https://doi.org/10.3390/rs12010086>

20. Maboudi, M., Amini, J., Malihi, S., Hahn, M.: Integrating fuzzy object based image analysis and ant colony optimization for road extraction from remotely sensed images. *ISPRS J. Photogramm. Remote. Sens.* **138**, 151–163 (2018). <https://doi.org/10.1016/j.isprsjprs.2017.11.014>
21. Manohar, N., Pranav, M.A., Aksha, S., Mytravarun, T.K.: Classification of satellite images. In: *ICTIS 2020. Smart Inn. Syst. Technol.* 703–713 (2020). https://doi.org/10.1007/978-981-15-7078-0_70
22. Matasci, G., Volpi, M., Kanevski, M., Bruzzone, L., Tuia, D.: Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **53**, 3550–3564 (2015)
23. Matikainen, L., Karila, K., Hyyppä, J., Litkey, P., Puttonen, E., Ahokas, E.: Object-based analysis of multispectral airborne laser scanner data for land cover classification and map updating. *ISPRS J. Photogramm. Remote Sens.* (128), 298–313 (2017). <https://doi.org/10.1016/j.isprsjprs.2017.04.005>
24. Mir, S.A., Padma, T.: Review about various satellite image segmentation. *Indonesian J. Electr. Eng. Comput. Sci.* **9**(3), 633–636 (2018). <https://doi.org/10.11591/ijeecs.v9.i3.pp633-636>
25. Mozgovoy, D., Hnatushenko, V., Vasyliiev, V.: Accuracy evaluation of automated object recognition using multispectral aerial images and neural network. In: *Proceedings of the SPIE 10806, Tenth International Conference on Digital Image Processing* (2018). <https://doi.org/10.1117/12.2502905>
26. Mozgovoy, D.K., Hnatushenko, V.V., Vasyliiev, V.V.: Automated recognition of vegetation and water bodies on the territory of megacities in satellite images of visible and IR bands. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **IV-3**, 167–172 (2018). <https://doi.org/10.5194/isprs-annals-IV-3-167-2018>
27. Mustafa, M.T., Hassoon, K.I., Hussain, H.M., Abd, M.H.: Using water indices (NDWI, MNDWI, NDMI, WRI and AWEI) to detect physical and chemical parameters by apply remote sensing and GIS techniques. *Int. J. Res.* **5**(10), 117–128 (2017). <https://doi.org/10.5281/zenodo.1040209>
28. Parvu, I.M., Picu, I.A.C., Dragomir, P., Poli, D.: Urban classification from aerial and satellite images. *J. Appl. Eng. Sci.* **10**, 163–172 (2020). <https://doi.org/10.2478/jaes-2020-0024>
29. Rad, A.M., Kreitler, J., Sadegh, M.: Augmented normalized difference water index for improved surface water monitoring. *Environ. Modell. Softw.* (140), 105030 (2021). <https://doi.org/10.1016/j.envsoft.2021.105030>
30. Saxena, J., Jain, A., Krishna, P.R.: Deep learning for satellite image reconstruction. In: *Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences*, pp. 569–577 (2022). https://doi.org/10.1007/978-981-16-5747-4_48
31. Shao, Q., Xu, C., Zhou, Yu., Dong, H.: Cast shadow detection based on the YCbCr color space and topological cuts. *J. Supercomput.* **76**(5), 3308–3326 (2018). <https://doi.org/10.1007/s11227-018-2558-4>
32. Shedlovska, Y.I., Hnatushenko, V.V.: A very high resolution satellite imagery classification algorithm. In: *Proceedings of the 2018 IEEE 38th International Conference on Electronics and Nanotechnology*, pp. 654–657 (2018). <https://doi.org/10.1109/elnano.2018.8477447>
33. Shedlovska, Y.I., Hnatushenko, V.V.: A shadow removal algorithm for remote sensing imagery. In: *Proceedings of IEEE 39th International Scientific and Technical Conference “Electronics and Nanotechnology”*, pp. 817–821 (2019). <https://doi.org/10.1109/elnano.2019.8783642>

34. Shelestov, A., et al.: Cloud approach to automated crop classification using Sentinel-1 imagery. *IEEE Trans. Big Data* **6**(3), 572–582 (2019). <https://doi.org/10.1109/tbdata.2019.2940237>
35. Singh, K.K., Pal, K., Nigam, M.J.: Shadow detection and removal from remote sensing images using NDI and morphological operators. *Int. J. Comput. Appl.* **42**(10), 37–40 (2012). <https://doi.org/10.5120/5731-7805>
36. Tamta, K., Bhadauria, H.S., Bhadauria, A.S.: Object-oriented approach of information extraction from high resolution satellite imagery. *IOSR J. Comput. Eng.* **17**(3), 47–52 (2015)
37. Xiao, P., Zhang, X., Zhang, H., Hu, R., Feng, X.: Multiscale optimized segmentation of urban green cover in high resolution remote sensing image. *Remote Sens.* (10), 1813 (20 pages) (2018). <https://doi.org/10.3390/rs10111813>
38. Xiaoxiao, L., Wenwen, L., Middel, A., Harlan, S.L., Brazel, A.J., Turner, B.L.: Remote sensing of the surface urban heat island and land architecture in Phoenix, Arizona: combined effects of land composition and configuration and cadastral-demographic-economic factors. *Remote Sens. Environ.* **174**, 233–243 (2016). <https://doi.org/10.1016/j.rse.2015.12.022>
39. Xue, J., Su, B.: Significant remote sensing vegetation indices: a review of developments and applications. *J. Sens.* 1353691 (17 pages) (2017). <https://doi.org/10.1155/2017/1353691>
40. Zhou, Y., Li, J., Feng, L., Zhang, X., Hu, X.: Adaptive scale selection for multiscale segmentation of satellite images. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **10**(8), 3641–3651 (2017). <https://doi.org/10.1109/JSTARS.2017.2693993>



CTrace: Language for Definition of Epidemiological Models with Contact-Tracing Transmission

Vladyslav Sarnatskyi^(✉)  and Igor Baklan 

National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”,
Kyiv, Ukraine

vladysarn@protonmail.com, iaa@ukr.net

Abstract. In this work, we present our results in development of a domain-specific programming language for an agent-based infectious disease spread modeling with contact tracing transmission. We performed an analysis of methods and tools, accessible to a scientific community, which showed significant problems of currently developed approaches. Some methods provide significant performance, sacrificing ease-of-use, others offer simple programming interface but suffer with performance drawbacks. To address both of these issues, we designed a domain-specific programming language with simple and yet powerful syntax and minimal amount of boilerplate code. The choice of the underlying infectious disease spread model is based on a growing interest in agent-based models, which are becoming more popular due to higher simulation accuracy compared to classical compartmental models. At its core, each agent has its personal user-defined daily schedule, which dictates where the agent is situated at each simulation step based on numerous parameters, such as simulation “day”, agent’s inner state, inner states of linked to this agent compartments, which can be, for example, its house, place of work, school, etc. The underlying language’s engine uses this schedule to check, which agent pair were in contact with each other and where at each simulation step. The infectious disease, defined by a set of rules, which control how do these contacts affect inner states of both agents and how disease progresses in context of a single infected agent. We designed a compiler with Python as intermediate language for a simple programming interface and Numba support for machine language translation. Performed tests showed, that the language is capable of describing complex epidemiological models and the compiler is able to generate machine code efficiently enough to run simulations with medium number of entities. The main problem of a developed approach is the quadratic nature of algorithmic complexity of a contact tracing transmission process. We left optimization of it as a topic for a future research.

Keywords: Modeling · Epidemiology · Agent-based models · Infectious disease spread modeling · Individual-based models · IBMs

1 Introduction

A global pandemic of COVID-19 caused by a SARS-CoV-2 virus showed, that the absence of optimal strategy in quarantine regulations can have significant impact on both epidemiological and economical state of society. The development of such strategies heavily relies on accurate pandemic models, which are developed by epidemiological research community. These models can be difficult to implement, as the development of them requires from a researcher not only in-depth knowledge of epidemical modelling, but also high programming and computer science skills. These skills are necessary, as they enable the developer to manage computational resources more efficiently and, thus, increase the number of modeled entities together with overall model accuracy.

With the significant increase in available computational resources, the popularity of so-called individual-based models (IBMs) also increased. Such models in contrast with classical compartmental models do not model the dynamics of groups of agents (compartments), but rely on more precise modeling of each individual separately. This type of models not only shows increased model accuracy, but also ability to capture complicated stochastic processes, inherent for the real world. The downside of such models is their significantly larger computational complexity, which require even higher resource management skills from a developer.

Even though it's not possible to develop disease models without knowledge in epidemiology, we can at least lower the requirement in programming skills by developing an epidemiological modeling platform, where all resource management is hidden from a user. Such platform can be implemented as compiled domain-oriented programming language with high abstraction level. The development of such language is a main topic of this research.

The main contributions of this paper are summarized as follows:

1. The paper investigates approaches to infection disease modeling from the perspective of both model types and software tools, outlining common advantages and disadvantages of them;
2. The mathematical background of an agent-based epidemiological model is presented together with corresponding modeling language formal grammar and examples of use;
3. Two simple epidemiological models based on European demographic data were implemented using developed modeling language, which showed the ability of it to be used in the case of complex models;
4. Several performance tests were conducted, which showed near to optimal performance of generated machine code and tolerable space complexity. However, the quadratic nature of model's step algorithms limits the use of this approach in large models. The solution to this problem was left for a future research.

The rest of the paper is structured in a following way. Section 2 outlines the main problem of this paper. Section 3 reviews current state of research in the field of epidemiological modeling. In Sect. 4 the definition of a mathematical background of a proposed agent-based epidemiological model in presented. Section 5 describes

the main structures of the developed modeling language and the intrinsics of a corresponding translator. Section 6 presents the results of conducted experiments on the ability of developed approach to be used in the development of a complex models, its time and space algorithmic complexity. Section 7 summarizes the research results and indicates the direction for future work. In Appendix A the formal grammar of the developed language is presented.

The aim of this research is to design a mathematical background of an agent-based epidemiological model with contact tracing transmission and a formal grammar of a corresponding domain-oriented language for its software implementation; develop a compiler for it; perform experiments to evaluate algorithmic complexity of compiled models.

2 Literature Review

A major type of research in epidemiological modeling is related to so-called compartmental models. These models divide a modelled population into several groups, defined by a number of members as a function of time. These functions relate to each other through a set of differential equations. The simplest of these models is SIR model, described in [5]. This model considers three population groups: susceptible (S), infectious (I) and recovered (R). Susceptible individuals are those who wasn't infected yet and can be infected by individuals from an infectious group. Recovered individuals were infected in the past and cannot be infected in the future because of developed immunity or death. The other type, called SIS, can model a disease, which can reinfect previously infected individuals. According to it, there is no recovered group, and infectious individuals are returned to the susceptible group after recovering. The SEIR model [6] adds a new compartment of exposed (E) individuals, which were exposed to a disease, but shown no symptoms. These individuals are often having lower transmission rate than infectious, sometimes even being unable to infect other individuals. There is a modification to SEIR model, namely SEIRS model [6], which is extended by an ability to model temporary immunity to a modeled disease. These models, when calibrated properly on a real data, can successfully model real diseases [2, 8, 12].

Individual-based models, describing each member of a population separately, are becoming more popular recently with the significant increase of available computational resources. Success of these models is caused by an ability to model complex stochastic interactions of a real world [4, 11]. For example, authors of [7] developed a complex model of an infectious disease spread in Europe by simulating both intra- and intercity movements of agents. The results showed, that it can model complex patterns of disease spread in European countries if properly initialized with real sociodemographic data. Also, it was shown, that this sociodemographic data has a significant impact on overall dynamics of disease spread.

To our knowledge, there are not many attempts to develop a modeling language for epidemiology. The only attempt, relying on a model of compartmental

type, that is presented to the public is Kendrick modelling platform [9]. The main part of this platform is Kendrick, which is a domain-specific modeling language, which can be used to define an arbitrary epidemiological model of a compartment type. There are several tools and languages dedicated to development of agent-based models. One of them is Swarm [1] - a library which offers some tools for implementing of such models. Despite its flexibility, Swarm requires from a user advanced knowledge of Java language programming skills, which limits its usage significantly among researchers in epidemiology. The ideological successor of Swarm is Ascape [10] library, which slightly simplifies programming interface. NetLogo [14] is both a complete software solution and a domain-oriented programming language dedicated to development and analysis of general-purpose agent-based models. It offers a simple user and programming interface, but suffers from a set of disadvantages. One of them is the design limitation of this particular solution which is built-in assumption, that each agent “lives” on a rectangular grid and is able to make contact with agents in adjacent grid tiles. This limitation can be avoided in a hacky way, which, we assume, can result in some performance issues. Another problem, which is not limited to NetLogo, but can be related to other general-purpose modeling languages, is the performance. The authors of [3] compared the performance of an epidemiological model, developed using NetLogo, with one, implemented using Oracle DBMS. It was found, that simulation step time of latter one was 5–100 smaller.

From this knowledge, we assume that the development of a domain-oriented modeling language targeted particularly on agent-based epidemiological models will be a great scientific contribution and a useful tool for a future research.

3 Method

3.1 Contract Tracing Model Overview

The underlying model consists of the following major objects:

- Infection. Models certain disease. Consists of infection states, transmission rules that define probability of agent’s infection state transition when encountered other infected agent and progress rule that define infection state transitions of single agent with time;
- Agents - play the role of humans in the simulated environment. Each has its infection state which is defined by modeling disease and daily schedule that defines agent’s movements in the simulated environment during simulation step based on its infection state and user-defined parameters;
- Compartments - act as geographical places and can be visited by agents during simulation step. Compartments can be of different types (common examples are: workplaces, schools, households).

Both agents and compartments can have a set of inner parameters and rules of their evolution, which can be used during validation of infection rules. These parameters can represent sociodemographic data, which was shown to be important during disease modeling [7].

3.2 Model Definition

To define a model with contact-tracing transmission, the following definitions were declared:

- S is the set of infection states. For example, $S = \{“S”, “E”, “I”, “R”\}$ for a classical SEIR model;
- P is the set of agents (individuals);
- C is the set of compartments;
- $T \subset \mathbb{N}$ is the set of simulation steps. Can be interpreted as days;
- $IS : P \times T \rightarrow S$ returns the infection state of agent $p \in P$ at simulation step $t \in T$;
- $H = \overline{1..n}$ is the set of time steps in each simulation step. Controls the granularity of simulation of each simulation step. For example, when $n = 24$ each time step can be interpreted as hour;
- $SC : T \times H \times P \rightarrow C$ is the schedule function, which defines daily schedule for each agent during simulation step. $s(t, h, p) = c$ means that at simulation step t , at time step h agent p was located in compartment c ;
- R^{prog} is the set of infection progress rules. Note, superscript $*^{prog}$ doesn't define power operation here, but serves as a part of notation. Each rule $r \in R^{prog}$ defined as a tuple (s_1, s_2, d, f) , $s_1 \in S, s_2 \in S, d \in \mathbb{N} \cup \{0\}, f : P \rightarrow [0, 1]$ describes the probability $P(IS(p, t + 1) = s_1 | IS(p, t) = s_2)$ of changing the agent's p infection state at the start of next simulation step $t + 1$ to s_1 when the infection state at current simulation step is s_2 and last state change was d simulation steps ago;
- R^{tran} is the set of infection transmission rules. Note, superscript $*^{tran}$ doesn't define power operation here, but serves as a part of notation. Each rule $r \in R^{tran}$ defined as a tuple (s_1, s_2, s_3, f) , $s_1 \in S, s_2 \in S, s_3 \in S, f : P^2 \times C \rightarrow [0, 1]$ describes the probability $P(IS(r, t + 1) = s_1 | IS(r, t) = s_2, IS(d, t) = s_3, SC(t, h, r) = c, h \in H)$ of changing agent's infection state at next simulation step $t + 1$ to s_1 when infection state at current simulation step is s_2 after contacting with donor with infection state s_3 in compartment c .

When defining an epidemiological model, it's common to define special infection state types: susceptible and infectious (for compartmental models, these will be compartment types). These state types should not be misinterpreted as S and I state of a SEIR-like models. Even though, S belongs to a set of susceptible states, one can define multiple of them.

As the described model uses a rule-based system to control disease spread, we don't require from a user a separate definition on such states. Instead, these states are derived automatically from a set of transmission rules:

- $S^s \subset S$ is the set of infection states, which make the agent susceptible to infection. Defined as $\forall s, s \in S^s \Leftrightarrow \exists (s_1, s_2, s_3, f) \in R^{tran}, s_2 = s$;
- $S^i \subset S$ is the set of infection states, which make agent infectious for other agents. Defined as: $\forall s, s \in S^i \Leftrightarrow \exists (s_1, s_2, s_3, f) \in R^{tran}, s_3 = s$;

Note, superscripts in S^s and S^i here don't define power operation, but serve as a part of notation.

The model's computational performance can be significantly increased by skipping agents with infection state which belongs neither to susceptible set, nor infectious during simulation step.

Based on this notation, the simulation algorithm it presented in Algorithms 1 and 2.

Algorithm 1: Infection spread algorithm

```

t ← current time step;
// loop over recepients
for r ∈ P do
  // skip agents with non-susceptible state
  if  $IS(r, t) \notin S^s$  then
    ⊥ continue
  // loop over donors
  for  $d \in P \setminus \{r\}$  do
    // skip agents with non-infectious state
    if  $IS(d, t) \notin S^i$  then
      ⊥ continue;
    for h ∈ H do
      // skip if donor and recepiet were at different
      compartments at step h
      if  $SC(t + 1, h, r) \neq SC(t, h, d)$  then
        ⊥ continue
       $IS(r, t + 1) \leftarrow s$  with probability  $f(d, r, SC(t, h, r))$  where
       $(s, IS(r, t), IS(d, t), f) \in R^{tran}$ 
      if  $IS(r, t + 1) \neq IS(r, t)$  then
        ⊥ break
  
```

Algorithm 2: Infection progress algorithm

```

t ← current time step;
// loop over agents
for p ∈ P do
   $IS(p, t + 1) \leftarrow s$  with probability  $f(p)$  where  $(s, IS(p, t), d, f) \in R^{prog}$  where
  ⊥ d - simulation steps since last infection state change for agent p
  
```

4 Modeling Language

Here, we describe developed language architecture and design principles.

4.1 Translator Intrinsic

We developed a compiler for the language, which has Python as its intermediate language. To ensure high computational efficiency needed to run models with large number of agents, all computations are done using Numba library [13], which performs JIT-compilation of a program from intermediate language to machine code. To reduce overhead from memory allocations, all variables are allocated in a static buffer, which size is enough to perform all necessary computations.

4.2 Expressions

The language supports common mathematical operators: $+$, $-$, $*$, $/$, $\hat{}$; which can use literals, variables or other expressions as operands:

```
a = 1+3;
b = a*a;
c = (b+a)*b;
```

A special case when one or more operands are sampled from some probability distributions has to be considered separately. To denote this sampling operation, the operator “ \sim ” is used. For example, in the following case:

```
a ~ D;
b = a + 1;
```

the value **a** is sampled from distribution **D** and the value **b** will be sampled from the respective distribution so that **b** will always be greater than **a** by 1 and recomputed at each evaluation of expression, containing **b**. When multiple instances of the same random variable appear in a single expression, its value sampled only once. For example, in the following case, value of **b** will always be equal to 1;

```
a ~ D;
b = (a + a) == (2 * a);
```

4.3 Parameters

Global numeric variables we interpret as model’s parameters. Their values can be set from ordinary expressions:

```
PI = 3.14;
PI_SQUARED_PLUS_ONE = PI^2+1;
```

or sampled from built-in or user-defined probability distributions:

```
sampled ~ D;
```

4.4 Distributions

The language allows one to define probability distributions of two possible types: regular and distributions over structure array. A regular distribution can be defined by specifying its inverse cumulative probability density function. For example:

```
D_coin {
    icdf(p) = p < 0.5;
}
```

will define probability distribution with outcome equal to zero or one with equal probability. The same probability distribution can be defined by directly specifying outcome values and probabilities:

```
D_coin {
    P(0) = 0.5;
    P(1) = 0.5;
}
```

Both values and probabilities can be dependent on external static variables. For example:

```
prob = 0.5;
one = 1
D_coin = {
    P(0) = prob;
    P(one) = 1 - prob;
}
```

Probability distribution definition can specify local parameters to this distribution:

```
Uniform(a, b) = {
    icdf(p) = a + (b - a) * p;
}
```

which have to be populated during sampling operation:

```
sampled_from_u_0_1 ~ Uniform(0, 1);
sampled_from_u_1_5 ~ Uniform(1, 5);
```

This example will define distribution **Uniform** with outcome sampled from $U(a, b)$ and two variables, sampled from $U(0, 1)$ and $U(1, 5)$.

The language supports another kind of regular probability distribution, which is a mixture of several uniform probability distribution. For example:

```
D_mixture {
    P([0, 1]) = 0.5;
    P([5, 10]) = 0.5;
}
```

will declare a mixture distribution consisting of two equiprobable uniform distributions: $U(0, 1), U(5, 10)$.

Distribution over a structure array is a type of distribution which has structure instance (Sect. 4.6) as an outcome. For example:

```
House [100] {
    size ~ D_house_size;
}
D_house {
    prob(h) = h.size;
}
```

Here, distribution **D_house** will output an instance of **House** with probability proportional to the value of **size**.

4.5 Functions

All user-defined functions can have single or multiple numeric arguments, must return a single numeric value, and cannot change the value of any global variable. For example:

```
f(x) = x ^ 2;
g(x, y) = x + f(y);
```

4.6 Arrays Structure

Structure arrays are an array of structures with constant, known at compile time length. Each structure member can be either numerical value, function, reference to another structure or transition.

Structure definition begins with array length specification followed by a set of structure members:

```
Person [100] {
    age = 25;
}
```

Will declare an array of structures **Person**, each having a member **age** with value 25. Like a normal parameter declaration, structure member can be sampled from some distribution:

```
Person [100] {
    age ~ D_age(some_param);
}
```

where **D** is some previously defined distribution with parameter *some_param*.

Each structure member declaration can reference not only globally defined variables, but also structure members defined before. For example:

```
Person [100] {
    age ~ D_age(some_parameter);
    is_adult() = age >= 18;
}
```

will declare structure with method, *is_adult* which will return 1 if *age* \geq 18 and 0 otherwise.

A structure can store a reference to another structure, sampled from a probability distribution, with full access to its members. For example:

```
School [10] {
    size ~ D_school_size ();
    opened = 1;
}
D_school {
    prob(s) = size;
}
Person [100] {
    school ~ D_school;
    age ~ D_age(some_parameter);
    is_child () = age < 18;
    visits_school () = is_child () *
    school.opened;
}
```

Here, the method's *visits_school* evaluation will be dependent on the value of the member *opened* of the referenced element of *School*.

Structure transitions are update rules which are applied to specified structure parameters at each simulation step. For example:

```
Person [100] {
    age ~ D_age(some_parameter);
    age -> age + 1;
}
```

will define an array of *Person*, whose members *age* are incremented by 1 at each simulation step.

4.7 Compartments and Agents

A special case of structure declaration is a declaration of compartments and agents. Compartment declaration starts with a specification string *\$compartment*, followed by a regular structure declaration. This specification allows the compiler to track which structures represent compartments and allocate them separately. Agent declaration starts with a specification string *\$agent* followed by a regular structure definition. The agent has to be defined exactly once and have a special method *schedule(h)*, which returns the agent's location at a time step *h*. For example, the following definition will model an environment where each agent stays at home during an entire simulation step.

```
$compartment
House [50] {
    size ~ D_size;
}
D_house {
```

```

        prob(h) = h.size;
    }
    Person [100] {
        house ~ D_house;
        schedule(h) = house;
    }

```

4.8 Disease Model

Disease model declaration consists of three parts: definition of states, progress rules and transitions.

Definition of disease states is a comma-separated list of state names in any order. For example:

```
S, E, I, R;
```

Progress rules are defined by direct specification of each member of $r \in R^{prog}$. For example:

```
P(E->I, 14) = 0.5;
```

will specify a rule, following which an infection will progress from state E to state I 14 time steps after last state change, i.e. when state became equal to E . Both probability value and time step delay can be specified by an expression, in which case, variable scope is extended by references to agent fields. This means, that infection progress can depend on agent's inner parameters. For example:

```

$agent
Person [100] {
    age ~ D_age;
}
...
P(E->I, 14) = 0.5 + age / 200;
...

```

Similarly to progress rules, transmission rules are also declared by specification of members of $r \in R^{tran}$. For example:

```
P(S->E|I) = 0.5;
```

will declare a rule, stating that a probability of changing a recipient's infection state from S to E when contacted with a donor with state I is equal 0.5. Transmission probability can be specified by an expression, in which case variable scope is extended by current recipient and donor references, available by identifiers d and r respectively. For example:

```

$agent
Person [100] {
    vaccinated ~ D_vaccinated;
}
...
P(S->E|I) = 0.5 * (1 - r.vaccinated);
...

```

This rule’s transmission probability will equal 0.5 if recipient is not vaccinated and 0 if it is.

Following example is a declaration of classical SEIR model of some imaginary disease with susceptible → exposed transmission:

```

Disease {
  S, E, I, R;

  P(E->I, 14) = 1.0;
  P(I->R, 14) = 1.0;

  P(S->E | I) = 0.5;
}
    
```

5 Experiment, Results and Discussion

In order to test the ability of the developed language to model complex environments, we implemented a contact tracing model based on the work of [13]. We model a city where agents have households and places of work based on their age and visit them according to their daily schedule. All probability distributions such as agents’ ages, households sizes, etc., are based on real data of European countries, gathered from EUROSTAT [13]. Age distribution was approximated using Gompertz distribution, fitted on real data. The dynamics of a disease is shown on Fig. 1. Full listing of model description is listed below:

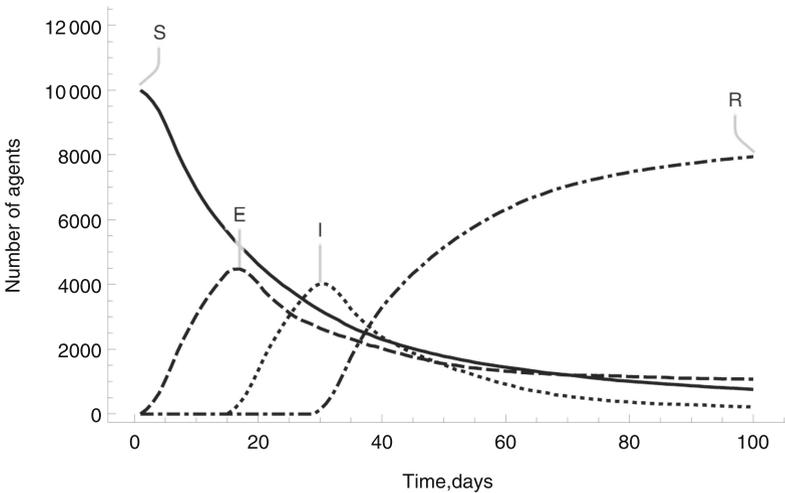


Fig. 1. Dynamics of disease states among agents in a simulated environment.

```

gompertz(c, loc, scale) {
    icdf(p) = ln(1-ln(1-p)/c)*scale+loc;
}
gaussian(x, mu, sigma) =
exp(-1 * ((x - mu) / sigma) ^ 2);
U(low, high) {
    icdf(p) = low + p * (high - low);
}

P_has_work(age) {
    icdf(p) = p < gaussian(
        age,
        41.56060857,
        21.78168914
    );
}

D_work_size {
    P(5, 10) = 0.46;
    P(10, 25) = 0.32;
    P(25, 50) = 0.125;
    P(50, 100) = 0.05;
    P(100, 200) = 0.02;
    P(200, 500) = 0.015;
    P(500, 1000) = 0.01;
}

D_closed_init {
    P(0) = 1;
    P(1) = 0;
}

D_house_size {
    P(2 ) = 0.571033;
    P(3 ) = 0.222209;
    P(4 ) = 0.156175;
    P(5 ) = 0.039507;
    P(6 ) = 0.008248;
    P(7 ) = 0.001891;
    P(8 ) = 0.000567;
    P(9 ) = 0.000209;
    P(10) = 0.000085;
}

$compartment
Work[300] {
    size ~ D_work_size;
    closed ~ D_closed_init;
}

P_workplace(Work) {
    prob(w) = size;
}

```

```

$compartment
Household[5000] {
    size ~ D_house_size;
    closed ~ D_closed_init;
}
P_household(Household) {
    prob(h) = size;
}

$agent
Person[10000] {
    age ~ gompertz(
        0.20510679560362763,
        -1.82765672236963,
        29.342139434877467
    );
    work ~ P_workplace;
    has_work ~ P_has_work(age);
    house ~ P_household;

    schedule(h) = work.closed * house +
        (1 - work.closed) * (work * (h > 7) +
        house * (h <= 7));
}
Flu {
    S, E, I, R;

    P(S -> E | I) = 0.0005;
    P(S -> E | E) = 0.0002;

    P(S -> E, 1) = 0.05;
    P(E -> I, 14) = 1;
    P(I -> R, 14) = 1;
}

```

By adding the following disease progress rule, we can simulate lost of immunity after 30 days:

$$P(R \rightarrow S, 30) = 1;$$

This will result in a disease dynamics depicted in Fig. 2.

To measure computational performance of the developed model, additional experiments were conducted. As the main bottleneck of provided model is *infection-spread* algorithm (which has quadratic complexity), we measured time needed to compute a single step of a model in the worst-case scenario. This scenario is defined as maximizing the number of contacts between agents. As we filter out non-susceptible and non-infectious agents, worst-case scenario would occur if part of the population would be infectious and other - susceptible. If we define number of susceptible agents as N_s and total population number as N ,

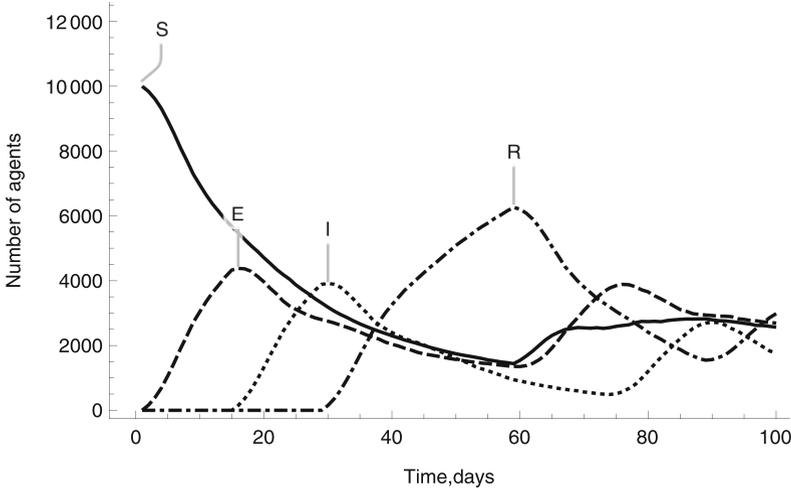


Fig. 2. Dynamics of disease states among agents with temporary immunity in a simulated environment.

worst-case number of susceptible agents can be defined as:

$$N_s = \operatorname{argmax}_x(x(N - x)) = \frac{N}{2} \quad (1)$$

In other words, average step time can be bounded from above by a step time of a state, where half of agents are susceptible and other - infectious. We performed a number of measurement for models with different number of agents, which infection state is randomly assigned to S or I with equal probability before each. Averages of 100 runs for each model are presented in Fig. 3¹.

If fitted to this data, polynomial regression with highest power equal to two has $R^2 > 0.99$, which confirms quadratic nature of algorithm's complexity. Such performance makes the whole approach somewhat limited, as it will be non-practical to run large models using it. For example, step time of a simulation with 10^6 agents would take approximately 45 min (according to the model). But, there are some optimizations can be made. If we assume, there aren't any rule in R^{tran} , which transmission probability function depends on recipient's inner state (except for its infection state), *infection.spread* algorithm can be reworked to have a linear complexity. This problem we leave for a future research.

A problem could arise from the choice of intermediate language. Python is known to be an inefficient language, compared to C/C++ or Rust. To prove that usage of JIT-compilation library Numba is enough to ensure good performance, developed compiler functionality was extended to be able to generate raw Python as well as Rust code. The same models were benchmarked in these environments. Results are shown in Fig. 4.

¹ All tests were done using a machine with Intel i7-8700K 3.7 GHz CPU.

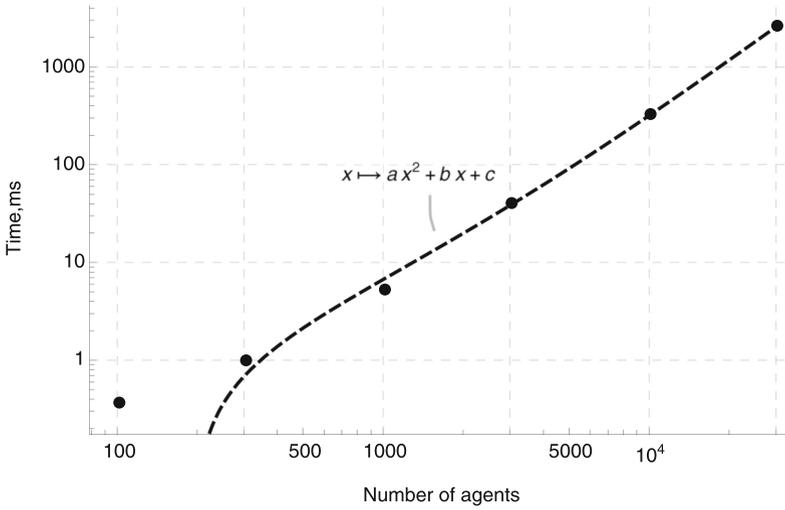


Fig. 3. Relation between step time and number of agents in a model. Regression parameters: $a = 2.67 \cdot 10^{-6}$, $b = 5.11 \cdot 10^{-3}$, $c = -1.09$.

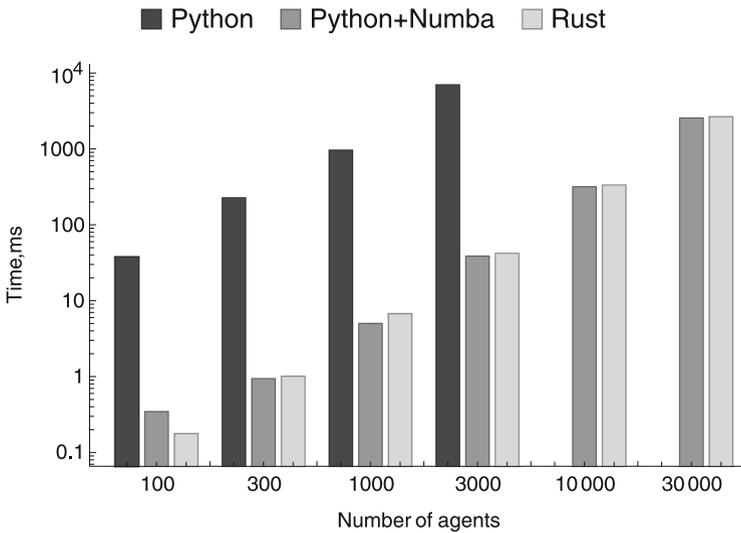


Fig. 4. Intermediate language performance for various model sizes.

Even though we can't guarantee that generated code is optimal for both languages, it is evident from the benchmark results that JIT-compiled version of Python has almost the same performance as Rust (Rust is 2.6% slower).

Remarkable performance gains can be seen when Python version is compared with Python+Numba: latter one is 260x faster².

From these results, we can conclude that using a JIT-compiled version of Python as an intermediate language offers not only easy-to-use programming interface, but also near to optimal performance.

Experiments were also carried out to estimate the space complexity of a developed model. As the compiler was designed to avoid memory allocations on model steps, we computed space complexity by evaluating buffer size of models with different number of agents. As it is evident from Fig. 5, developed model has linear space complexity as a function of number of agents ($R^2 = 1$).

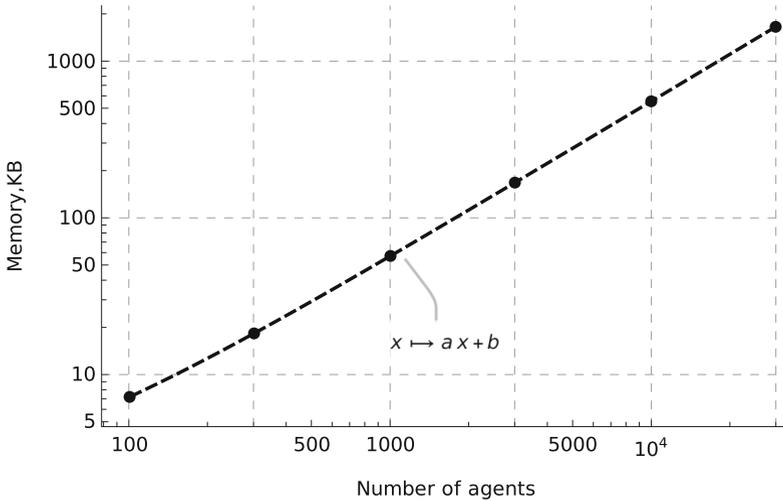


Fig. 5. Relation between buffer size and number of agents in a model. Regression parameters: $a = 5.52 \cdot 10^{-2}$, $b = 1.61$.

6 Conclusions

In this paper, a mathematical model for an agent-based epidemiologic model with contact-tracing transmission was proposed. This model consists of three major entities and entity groups: infection, which describes infection progress rules in the scope of a single agent, as well as spread rules between agents; agents, which play a role of creatures, susceptible to a modeled disease; compartments, which are places agents visit correspondingly to a daily schedule and meet each other. A formal grammar of a domain-specific programming language was designed

² Comparison was performed by evaluating the ratio between highest power coefficients of a fitted regression model.

based on this model and a corresponding compiler with JIT-compiled Python as an intermediate language and application programming interface of a generated artifact in Python language was implemented. Performed experiments showed the ability of a developed approach to implement complex epidemiological models, as well as near to optimal performance of a generated machine code. However, there is a problem with quadratic time complexity of simulation step algorithm, which limits the usage for models with large number of agents. The inspection of a possible solution was left for a future research.

A Appendix

Here, we describe a formal grammar of a developed language.

The set of terminal symbols is defined as: $\Sigma = \{0..9, a..z, A..Z, +, -, *, /, ' == , ! =, < >, <, >, < =, > =, =, (,), \{, \}, [,], ,, \$, - >, < -, \}$.

The set of non-terminal symbols is defined as: $N = \{ \text{"letter", "digit", "number", "id", "expression", "program", "distribution_declaration", "distribution_disc_declaration", "distribution_cont_declaration", "function_ declaration", "param_declaration", "param_sampling", "struct_declaration", "transition", "infection"} \}$.

The start symbol S is "program".

Production rules, presented using railroad diagrams, are defined in Figs. 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 and 21.

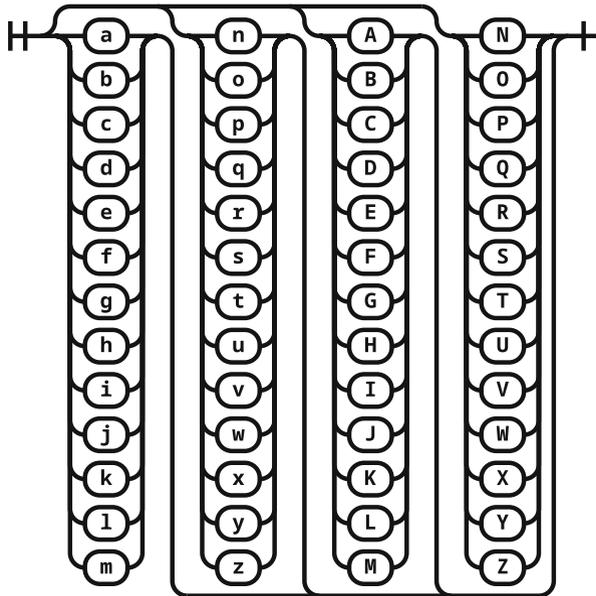


Fig. 6. Definition of *letter* non-terminal.

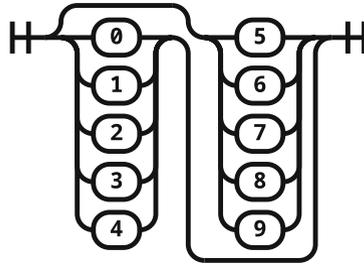


Fig. 7. Definition of *digit* non-terminal.

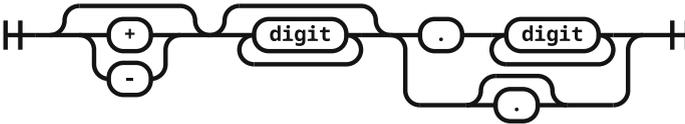


Fig. 8. Definition of *number* non-terminal.

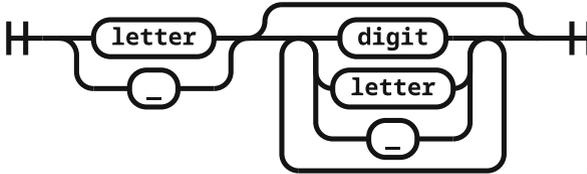


Fig. 9. Definition of *id* non-terminal.

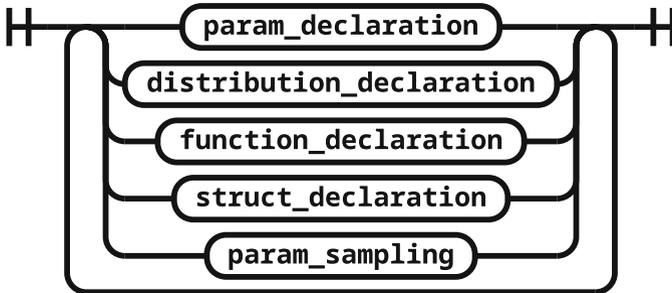


Fig. 10. Definition of *program* non-terminal.

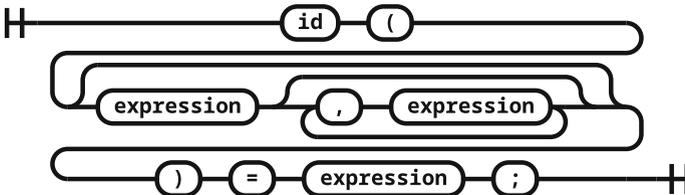


Fig. 11. Definition of *function_declaration* non-terminal.

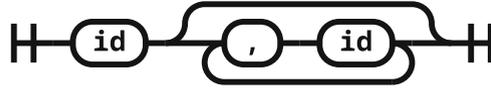


Fig. 12. Definition of *function_declaration_arguments* non-terminal.



Fig. 13. Definition of *param_declaration* non-terminal.



Fig. 14. Definition of *distribution_declaration* non-terminal.

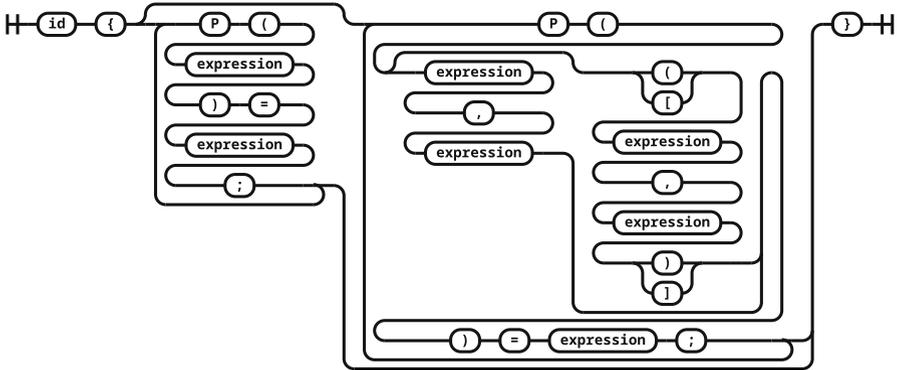


Fig. 15. Definition of *distribution_disc_declaration* non-terminal.

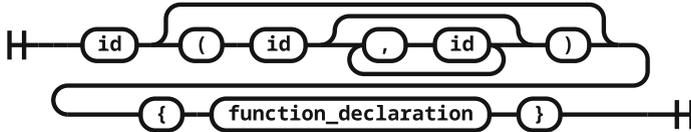


Fig. 16. Definition of *distribution_cont_declaration* non-terminal.

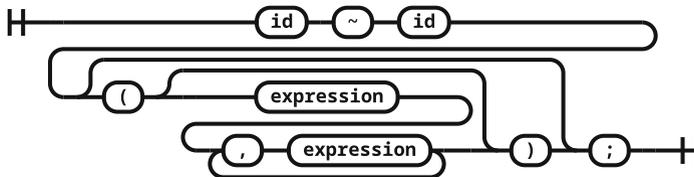


Fig. 17. Definition of *param_sampling* non-terminal.

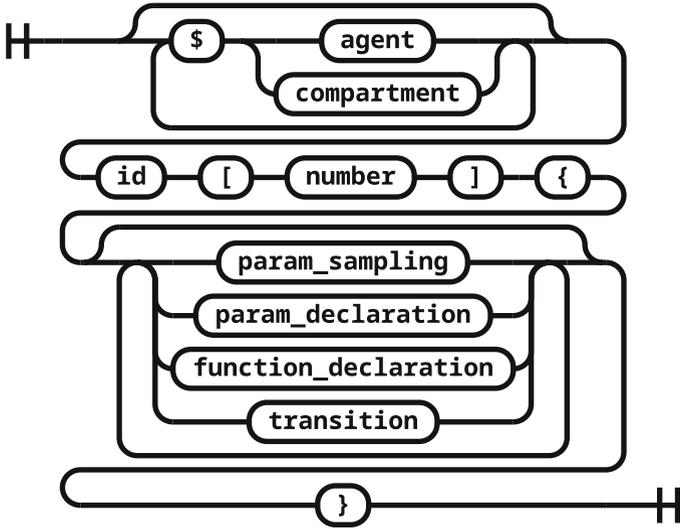


Fig. 18. Definition of *struct_declaration* non-terminal.

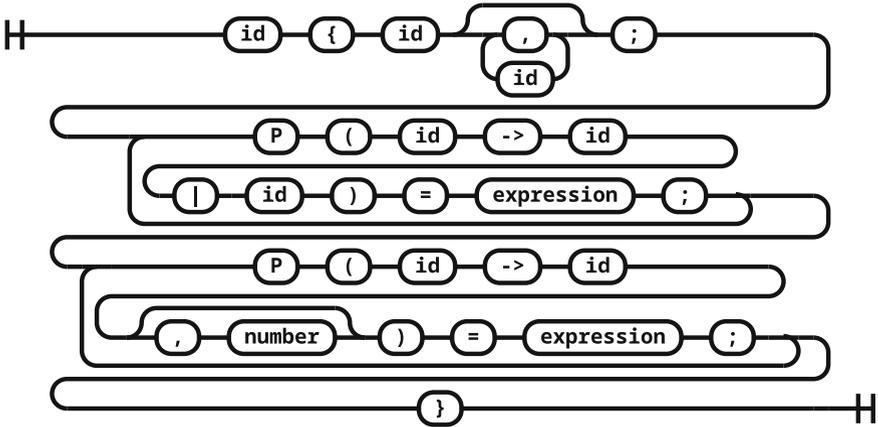


Fig. 19. Definition of *infection* non-terminal.

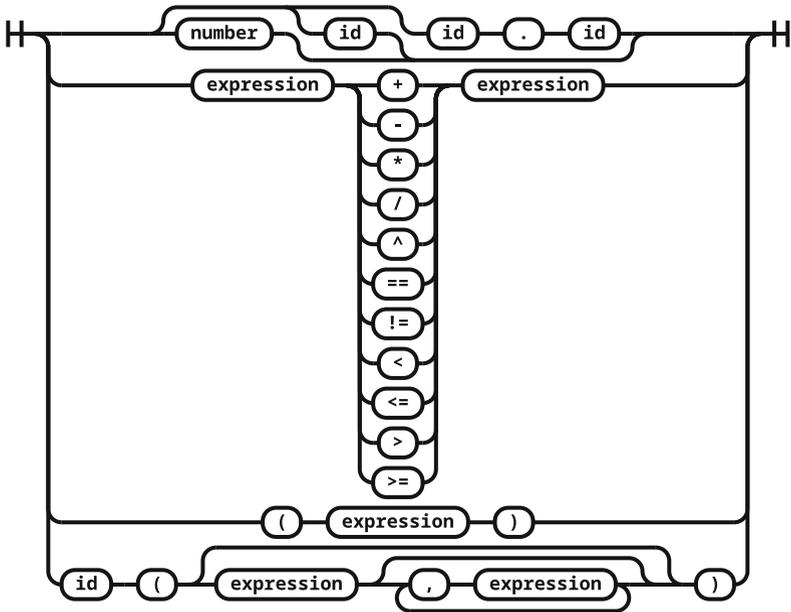


Fig. 20. Definition of *expression* non-terminal.

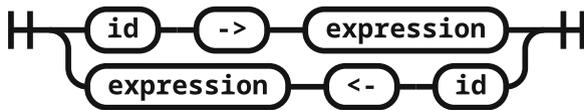


Fig. 21. Definition of *transition* non-terminal.

References

1. Swarm (2021). http://www.swarm.org/wiki/Main_Page
2. Alvarez, F.E., Argente, D., Lippi, F.: A simple planning problem for Covid-19 lockdown. Technical report, National Bureau of Economic Research (2020). <https://doi.org/10.3386/w26981>
3. Avilov, K., Solovey, O.Y.: Institute of Numerical Mathematics of Russian Academy Of Sciences Moscow Institute of Physics and Technology (State University)
4. Eisinger, D., Thulke, H.H.: Spatial pattern formation facilitates eradication of infectious diseases. *J. Appl. Ecol.* **45**(2), 415 (2008). <https://doi.org/10.1111/j.1365-2664.2007.01439.x>
5. Harko, T., Lobo, F.S., Mak, M.: Exact analytical solutions of the susceptible-infected-recovered (SIR) epidemic model and of the sir model with equal death and birth rates. *Appl. Math. Comput.* **236**, 184–194 (2014). <https://doi.org/10.1016/j.amc.2014.03.030>
6. Hethcote, H.W., Van den Driessche, P.: Some epidemiological models with non-linear incidence. *J. Math. Biol.* **29**(3), 271–287 (1991). <https://doi.org/10.1007/BF00160539>
7. Merler, S., Ajelli, M.: The role of population heterogeneity and human mobility in the spread of pandemic influenza. *Proc. Roy. Soc. B: Biol. Sci.* **277**(1681), 557–565 (2010). <https://doi.org/10.1098/rspb.2009.1605>
8. Ng, W.L.: To lockdown? When to peak? Will there be an end? A macroeconomic analysis on Covid-19 epidemic in the united states. *J. Macroecon.* **65**, 103230 (2020). <https://doi.org/10.1016/j.jmacro.2020.103230>
9. Papoulias, N., Stinckwich, S., Ziane, M., Roche, B., et al.: The Kendrick modelling platform: language abstractions and tools for epidemiology. *BMC Bioinform.* **20**(1), 1–13 (2019). <https://doi.org/10.1186/s12859-019-2843-0>
10. Patlolla, P., Gunupudi, V., Mikler, A.R., Jacob, R.T.: Agent-based simulation tools in computational epidemiology. In: Böhme, T., Larios Rosillo, V.M., Unger, H., Unger, H. (eds.) *IICS 2004*. LNCS, vol. 3473, pp. 212–223. Springer, Heidelberg (2006). https://doi.org/10.1007/11553762_21
11. Railsback, S.F., Grimm, V.: *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton University Press, Princeton (2019)
12. Rădulescu, A., Williams, C., Cavanagh, K.: Management strategies in a Seir-type model of Covid 19 community spread. *Sci. Rep.* **10**(1), 1–16 (2020). <https://doi.org/10.1038/s41598-020-77628-4>
13. Sun, W., Jiang, W., Trulls, E., Tagliasacchi, A., Yi, K.M.: ACNe: attentive context normalization for robust permutation-equivariant learning (2020). <https://doi.org/10.48550/arXiv.1907.02545>
14. Wilensky, U.: *Netlogo 6.2.2 user manual* (2022). <http://ccl.northwestern.edu/netlogo/docs/>



Optimization of Data Preprocessing Procedure in the Systems of High Dimensional Data Clustering

Maksym Korobchynskiy^(✉) , Myhailo Rudenko , Vladyslav Dereko ,
Oleksandr Kovtun , and Oleksandr Zaitsev 

Military Academy named after Eugene Bereznyak, Kyiv, Ukraine
{maks_kor,ruminik33,vladyslavderek}@ukr.net

Abstract. In this paper, we propose the technique of the optimal method choice of high dimensional data normalizing at the stage of data preprocessing procedure is performed. As well known, the qualitative carried out of the data preprocessing procedure significantly influences the further step of their processing such as classification, clustering, forecasting, etc. Within the framework of our research, we have used both the Shannon entropy and the relative ratio of Shannon entropy as the main criteria to evaluate the data normalizing quality. Before the apply the cluster analysis, we reduce the data dimensionality by using the principal component analysis. The obtained data clustering was performed using a fuzzy C-means clustering algorithm with an evaluation of the data clustering quality when using various methods of data normalizing. The analysis of the simulation results allows us to conclude that for this type of data (gene expression profiles) the decimal scaling method is optimal since the Shannon entropy of the investigated data achieves the minimal value in comparison with the use of other normalizing methods. Moreover, the relative ratio of Shannon entropy does not exceed the permissible norms during the data dimensionality reduction by applying the principal component analysis technique.

Keywords: High dimensional data clustering · Shannon entropy · Data preprocessing · Normalizing · Optimization · Principal component analysis

1 Introduction

At present, in many areas of scientific research, there is a necessity to develop models and systems for complex data clustering, the distinguishing feature of which is the high dimension of the feature space. Such data are various types of images, gene expression data, time series, etc. To the qualitative process of these

types of data considering the current problem (classification, clustering, forecasting, etc.), very important is the preprocessing stage, which can include: missing value handling, data normalizing, data filtering, and dimensionality reduction. One of the principal methods currently used to reduce the feature space dimensionality is the principal component analysis, which allows us to significantly reduce the number of features that characterize an object while retaining useful information about the object to the maximum. However, it should be noted, that the successful application of this technique limited its high sensitivity to data preprocessing methods. For example, in the case of cluster analysis implementation, the result of splitting objects into clusters can vary significantly depending on the applied combination of preprocessing methods. One of the important steps of data preprocessing is their normalizing, which results in the reduction of the initial data to the required range and distribution. Currently, there are a large number of different normalization methods focused on different types of data. As a result, an urgent problem arises in creating an effective technique for the reasonable choice of a method for complex high-dimensional data normalizing, which can allow us, with a further reduction in the dimension of the feature space, to maximize the effectivity of the further data processing of the investigated data taking into account the current problem.

2 Problem Statement

Let the initial data be presented as a matrix:

$$X = \{x_{ij}\}, i = \overline{1, n}, j = \overline{1, m} \quad (1)$$

where n and m are the number of rows (objects) and columns (objects' features) respectively.

The choice of a set of features is carried out as a result of a narrowing mapping:

$$\{X^m\} \xrightarrow{F} \{X^k\}, k < m \quad (2)$$

in which the extremum of a certain quality criterion $J_X(F)$ is reached. F in (2) is the transformation function of the set $\{X^m\}$ into set $\{X^k\}$, k in this instance is the dimension of new features space.

It should be noted that each mapping (2) is associated with some value of the criterion $J_X(F)$. The result of the obtained mappings is the function $g(J_X)$. The main problem in this case is to find such a mapping that achieves the extremum of the function $g(J_X)$. Within the framework of our research, to choose an optimal data normalization method, we apply, as the function $g(J_X)$, the Shannon entropy criterion, which is currently used successfully in various areas of the scientific research [9, 19, 22]:

$$H(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (3)$$

Entropy criterion values, in this case, are calculated for the original non-normalized signal, after normalization, and for the calculated principal components. In accordance with the principle of maximum entropy [20]: “For given knowledge in the form of constraints, there is only one distribution that satisfies these constraints, which can be chosen using a procedure that satisfies the “consistency axioms”. This unique distribution is determined by entropy maximization”. In accordance with the hereinbefore, the following experimental procedure is proposed:

- filtering of the initial data in order to minimize the studied data noise component;
- calculation of the Shannon entropy of the non-normalized data;
- data normalization by the apply of various normalization methods;
- calculation of the Shannon entropy of the normalized data;
- calculating the principal components of each matrix of the normalized data;
- calculation of the Shannon entropy of the obtained principal components;
- clustering a set of investigated objects using the fuzzy C-means clustering algorithm;
- obtained results analysis and interpretation.

3 Literature Review

Currently, a large number of cluster analysis algorithms have been developed and implemented in various areas of scientific research [2, 6, 14, 25]. Their principal differences from each other are the choice of how to form a cluster structure, the used proximity metrics, the presence or absence of self-organization, etc. The reviews [10, 23, 24] present a block chart of the step-by-step procedure for forming a cluster structure and a block chart of methods for forming a cluster structure based on the most common data clustering algorithms. The authors have shown that choice of the appropriate algorithm is determined by the type of the investigated data and the desired method of a cluster structure formation. The result of hierarchical clustering algorithms implementation is a dendrogram or a tree of decisions at different stages of the objects grouping. In this instance, it should be noted that various methods of the same algorithm implementation can generate different structures of dendrograms, which can affect the objectivity of the final decision regarding the cluster structure formation. Kohonen’s self-organizing maps [16, 21] and their logical continuation SOTA clustering algorithm (Self Organizing Tree Algorithm) [8, 13] to some extent solve the problem of cluster structure formation, but the result of their operation depends on the choice of the algorithm parameters by assessment of the appropriate clustering quality based on the analysis of quantitative clustering quality criteria, which complicates their successful application.

Recently, with the development of bioinformatics methods, databases of cells of biological objects have been created, which are characterized by high dimensionality (tens of thousands) and high noise levels that are determined by biological and technological factors which arise during both the preparation and

conducting experiments regarding the experimental data formation [15,18]. For this reason, there is a necessity to develop effective methods for this type of data, their preprocessing and reducing the dimensionality size without loss of significant information about the investigated objects. For this reason, there is a necessity to develop effective methods for this type of data, their preprocessing and reducing the dimensionality size without loss of significant information about the investigated objects based on the use of current achievements in data science field [1,3,4]. One of the important stages of data preprocessing is their normalization. In the general case, all normalization methods, focused on high-dimensional gene expression data, can be divided into two subgroups: methods that use the reference subset of the etalon object and methods that use the whole set of the investigated data. In the first case, the normalization methods are divided into linear and nonlinear ones. In the second case, can be used the method of cyclic local regression, the contrasts and quantile normalization methods. In [7], the authors present the results of the research regarding a comparative analysis of existing normalization methods using the data samples obtained based on the use of DNA microarray experiments. The authors have shown that methods that do not use a reference subset allow obtaining a better quality of data normalization. The quantile normalization method proved to be more effective in pairwise normalization. In addition, the method of quantile normalization proved to be the fastest of the methods used. However, it should be noted that despite the progress in this subject area, there are some unresolved or partially resolved issues.

The unsolved parts of the general problem include the lack of an effective technique for choosing the methods of both high dimensional gene expression data preprocessing and size dimensionality reduction when the data mining and/or machine learning models creation [3,17]. In this instance, the goal of the paper is to develop a technique for choosing the optimal normalization method at the stage of high-dimensional data of complex biological nature preprocessing with subsequent reduction of dimensional space and clustering of objects based on the use of Shannon entropy as the main criterion for assessing the quality of information processing.

4 Material and Methods

In accordance with the hereinbefore presented stepwise procedure of the simulation performing, the first stage is the experimental data filtering, the purpose of which is to minimize the noise component. The data filtering procedure was carried out using wavelet analysis [11]. Wavelets are spatially localized functions capable of tracking and properly processing the local features of the investigated data. The efficiency of wavelets applying for high-dimensional data preprocessing is determined by the nature of the objects' feature space distribution. For example, in the case of the analysis of DNA microarrays, the features are the gene expression values, while the genes on the microarray are distributed in such a way that different genes with different levels of expression will alternate with a

frequency significantly lower than the frequency of the background noise. In the case of DNA microarray data use, the level of gene expressions can be assessed by the light intensity at the appropriate point in the DNA microarray. During the wavelet decomposition procedure performing, its low-frequency component is extracted from the original signal, while the local features of the original data are preserved. The main idea of the data wavelet decomposition is as follows: the initial data space corresponding to different levels of gene expression is decomposed into a system of subspaces so that each subsequent subspace is nested in the previous one:

$$I \subset I_1 \subset I_2 \subset I_3 \subset \dots \subset I_n \tag{4}$$

The resulting subspaces must satisfy the following property:

$$\bigcap_{i=0}^n I_i = \{0\} \tag{5}$$

Each of the resulting subspaces is decomposed, in turn, using the wavelet function and the scaling function into spaces of approximating and detailing coefficients:

$$I_i = \{A_i, H_i, V_i, D_i\} \tag{6}$$

where: A is the matrix of approximating coefficients; H, V, D are the horizontal, vertical, and diagonal detailing coefficients respectively.

If the object is characterized by one vector of features, the result of the wavelet decomposition will be a vector of approximating and one vector of detailing coefficients at the appropriate level of wavelet decomposition. The approximating coefficients contain information about the low-frequency component of the signal, while the detailing coefficients contain information about the high-frequency component of the signal. In most cases, the useful component of the signal is contained in the low-frequency component, and the detailing coefficients contain information about the local features of the signal. The noise component also is contained in the high-frequency component, which is also included in the detailing coefficients. Thus, we can conclude that by processing the detailing coefficients in the appropriate way and restoring the signal from the approximating coefficients and the processed detailing coefficients, we obtain a matrix of features of the investigated objects with a minimum level of the noise component. This fact can improve the accuracy of the solution of appropriate problems.

In [5], the authors presented the results of the research focused on determining the type of wavelet, the level of wavelet decomposition, and the values of the thresholding parameter to process the detailing coefficients when the acoustic emission signal was processed. The entropy criterion was used as the main criterion for assessing the filtration procedure quality. Taking into account the results obtained in [14], when constructing the object clustering model at the filtering stage, the symplet-4 wavelet was used at the fourth level of wavelet decomposition with soft processing of detailing coefficients.

The purpose of the normalization step is to transform the empirical data to the same range and distribution, most often to normal. Within the framework of our research, we used the following data normalization methods:

- minimax normalization:

$$XN[i, j] = \frac{X[i, j] - X_{min}(i)}{X_{max}(i) - X_{min}(i)} \tag{7}$$

- decimal scaling:

$$XN[i, j] = \frac{X[i, j]}{10^k} \tag{8}$$

where parameter k is chosen so that the maximum value of the element in the array is less than 1;

- normalization with the use of standard deviation:

$$XN[i, j] = \frac{X[i, j] - \bar{X}(i)}{\sigma(i)} \tag{9}$$

- linear normalization method proposed by the developers of Affymetrix company in relation to the analysis of DNA microarrays:

$$XN[i, j] = \frac{\bar{X}_{ref}}{\bar{X}(i)} X[i, j] \tag{10}$$

where \bar{X}_{ref} is an average of expression values in the reference vector of gene expressions;

- non-linear normalization using logistic transfer function:

$$XN[i, j] = \frac{1}{1 + \exp(-\frac{X[i, j] - \bar{X}(i)}{\sigma(i)})} \tag{11}$$

- method of contrasts, in which it is assumed that between the vectors M and A there is a linear regression relationship. Vectors M and A in this case are formed according to the formulas:

$$M[i, j] = \log_2\left(\frac{X[i, j]}{X[ref, j]}\right) \tag{12}$$

$$A[i, j] = \log_2(X[i, j] - X[ref, j]) \tag{13}$$

The normalizing correction and normalized values are calculated as follows:

$$\delta M[i, j] = M[i, j] - \hat{M}[i, j] \tag{14}$$

$$XN[i, j] = 2^{\left(A[i, j] + \frac{\delta M[i, j]}{2}\right)} \tag{15}$$

- quantile normalization is based on the assumption that the features that characterize n -objects have the same data distributions. Then the graph of quantiles in n -dimensional space will be a line lying along the diagonal, the coordinates of which are represented by the vector: $(\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}})$. The normalized values of the features of the studied vectors, in this case, are the projections of the point of the n -dimensional graph of quantiles into this diagonal.

The assessment of the quality of the investigated objects features preprocessing was assessed by analyzing the nature of the Shannon entropy criterion variation. If by the current event we understand the value of the attribute j in the vector i $X[i, j]$, then the probability of this event occurring is $p(i, j) = X[i, j]^2$. Then, taking into account that as a measure of the amount of information in the studied vector, entropy is a dimensionless quantity, formula (3) for the i -th row can be represented as follows:

$$H_i = - \sum_{j=1}^n X[i, j]^2 \log_2(X[i, j]^2) \quad (16)$$

In accordance with the concept of connection between entropy and information regarding the state of the system, the maximum amount of information corresponds to the minimum of entropy. Then it is natural to assume that the most high-quality preprocessing of object features can be selected from the condition of minimum Shannon entropy, which corresponds to the maximum information about the investigated objects particularity, while in the process of performing the principal component analysis, the change of the Shannon entropy values should be minimal, which indicates a minimum loss of information during the data transformation.

5 Experiment, Results and Discussion

The simulation process was carried out based on the integrated use of the R and Python programming languages. The data wavelet filtering stage was carried out using tools of the Python program and, principal component analysis and data clustering with the calculation of Shannon entropy were done using tools of R software. The database of patients with leukaemia [12], which is an array in size of 72×7131 , was used as an experimental base for the research. Each line contains information about the level of gene expression of diseased cells of appropriate patients. The investigated objects are divided into two classes. One class represents samples of cells taken from patients with acute myeloid leukaemia, and other cells represent patients with acute lymphoblastic leukaemia. The C-means fuzzy clustering algorithm was chosen for the implementation of the data clustering stage. This choice is determined by the nature of the objects distribution in the feature space. Variations in the characteristics of the studied objects due to their natural features, do not allow us to unambiguously attribute the object

to one or another cluster. Only the statement about the degree of belonging of an object to one or another cluster makes sense. For this reason, fuzzy clustering methods are most appropriate for the analysis of this type of data, one of which is the C-means clustering algorithm.

The distribution chart of the percentage of correctly distributed objects, when various normalizing methods were used, is shown in Fig. 1. Figure 2 shows the charts of the Shannon entropy values distribution which were calculated for both the initial data and the data normalized by various methods. Figure 3 shows similar charts for the data, attributes of which are the calculated principal components.

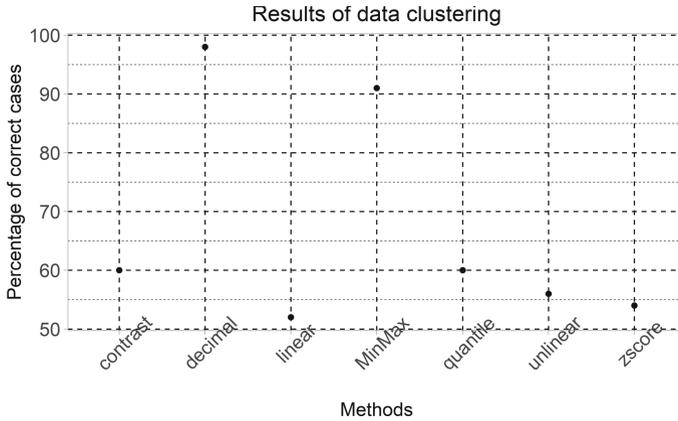


Fig. 1. The distribution diagram of the percentage of objects correctly distributed over clusters when the various normalization methods were used

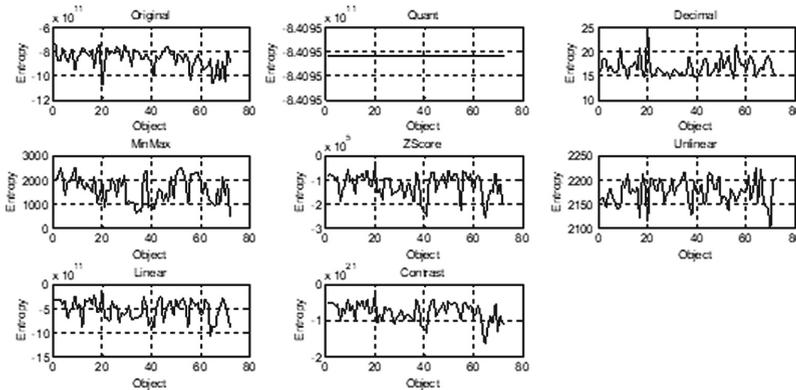


Fig. 2. Charts of Shannon entropy values of the original and normalized data by various normalizing methods

To assess the degree of Shannon entropy values change during the data transformation by calculating the principal components, the relative ratio of Shannon entropy was calculated when the data values were transformed:

$$\frac{dH_i}{H_{0i}} = \frac{|H_i - H_{0i}|}{H_{0i}} \tag{17}$$

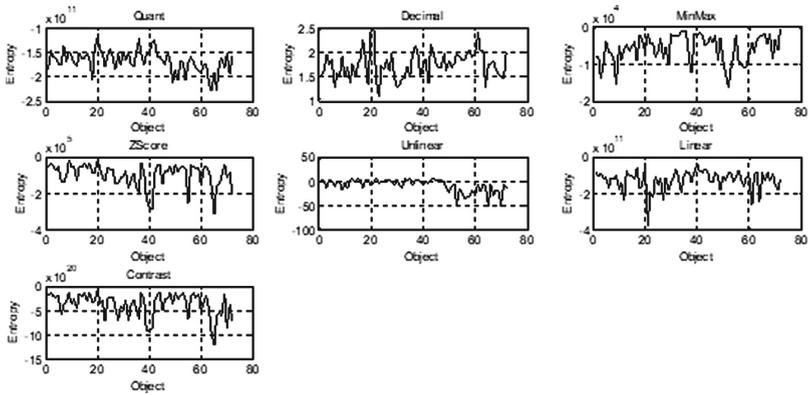


Fig. 3. Charts of Shannon entropy values of the data used the principal components as attributes

Charts of the Shannon entropies relative change calculated for both the normalized and transformed data when were applied various normalization methods are shown in Fig. 4.

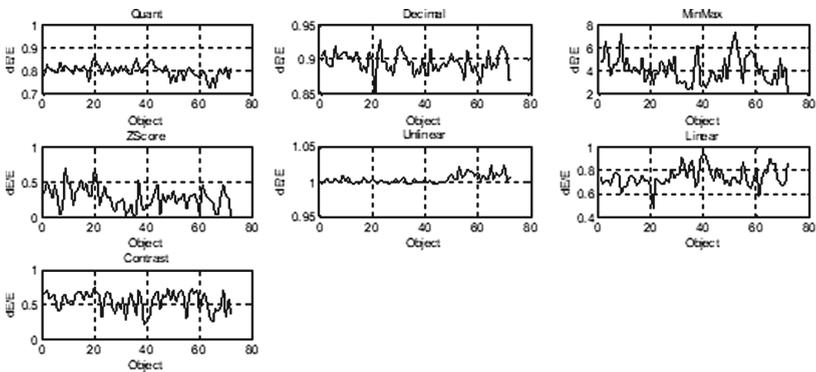


Fig. 4. Charts of the Shannon entropies relative change calculated for both the normalized and transformed data when were applied various normalization methods

An analysis of Fig. 2 allows us to conclude that the data normalized by the decimal-scaling, min-max and nonlinear methods have the least entropy. In this case, the decimal scaling method turns out to be more preferable in accordance with the entropy criterion. However, the absolute value of the entropy criterion is not objective, since the range of variation of the investigated objects feature values varies depending on the current normalization method, which directly affects the absolute value of the entropy criterion. The magnitude of the information loss due to the data transformation (calculation of the principal components) can be judged by the relative criterion of the Shannon entropy change. From the analysis of the Charts depicted in Figs. 2, 3 and 4 it follows that when using the methods of quantile normalization, decimal scaling, zscaling, linear normalization and normalization by the contrast method, approximately the same loss of information occurs. However, more detailed analysis of these two criteria values shows that the decimal scaling method is the most preferable for this type of data normalizing. When using the minimax and nonlinear normalization methods, a more significant increase of Shannon entropy values is observed, and hence a large loss of information regarding the studied object is performed, which is confirmed by the clustering quality shown in Fig. 1.

Figure 5 shows diagrams of the distribution of the objects into clusters when using decimal-scaling and MinMax normalizing methods.

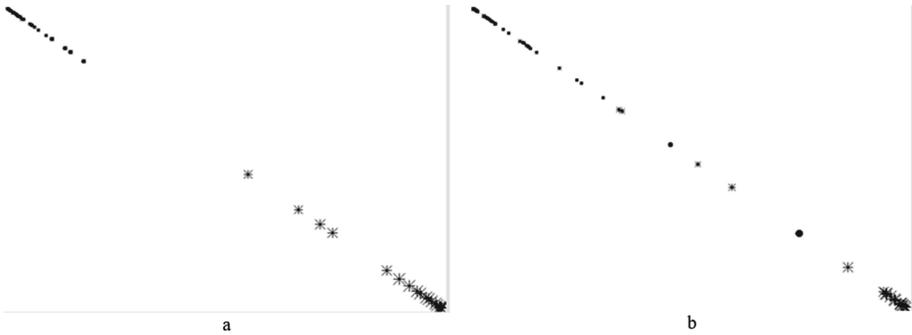


Fig. 5. Results of objects distribution into clusters when using: a) decimal-scaling normalizing method; b) MinMax normalizing method

An analysis of the objects distribution into clusters, shown in Fig. 5, confirms the obtained results concerning the higher efficiency of the decimal-scaling normalization method in comparison with other methods since the cluster structure in this instance (Fig. 5a) does not have any intersection, the objects are distributed into clusters in the most cases correctly.

6 Conclusions

The paper further developed the technique of using the Shannon Entropy criterion as an information criterion for assessing the amount of useful information in

the investigated data, a distinctive feature of which is the high dimension of the feature space. A comprehensive approach has been proposed for assessing the quality of information processing based on both the Shannon entropy criterion and the relative ratio of Shannon entropy during data transformation using the principal component analysis method. The performed investigation has shown that the principal component analysis method is highly sensitive to data normalizing methods at the stage of their preprocessing, as a result of which research on optimizing the use of high-dimensional data preprocessing methods in order to improve the quality of their further processing (clustering, classification, forecasting).

The practical interest is the use of the technique for processing DNA microarrays, the objects of which are the cells of a diseased organ, and the features are the genes expression values that determine the state of the corresponding cell. A particularity of the studied data is a high noise level and a high dimension of the feature space, which limits the use of traditional methods of this type of data processing. The research results have shown that for data of a biological nature, which are presented as a matrix of gene expression of diseased cells, the least loss of information is observed when using the decimal-scaling data normalization method. The number of objects correctly distributed into clusters, in this case, is 98%. The studies have also shown that in the case of the absence of a priori information about the object belonging to a particular cluster, the optimal combination of preprocessing methods can be determined based on the Shannon Entropy criterion. The value of this criterion for the most qualitatively preprocessed data is minimal, which corresponds to the maximum amount of useful information. In addition, the relative change of the Shannon entropy criterion values during signal transformation by the application of principal component analysis in the case of using the decimal-scaling method does not exceed the allowable value, which indicates a slight loss of information during the data transformation.

References

1. Andrunyk, V., Vasevych, A., Chyrun, L., et al.: Development of information system for aggregation and ranking of news taking into account the user needs. In: CEUR Workshop Proceedings, vol. 2604, pp. 1127–1171 (2020)
2. Babichev, S., Škvor, J.: Technique of gene expression profiles extraction based on the complex use of clustering and classification methods. *Diagnostics* **10**(8), 584 (2020). <https://doi.org/10.3390/diagnostics10080584>
3. Babichev, S., Lytvynenko, V., Skvor, J., Korobchynskiy, M., Voronenko, M.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction. In: Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, pp. 336–341 (2018). <https://doi.org/10.1109/DSMP.2018.8478452>
4. Babichev, S., Osypenko, V., Lytvynenko, V., et al.: Comparison analysis of biclustering algorithms with the use of artificial data and gene expression profiles. In: 2018 IEEE 38th International Conference on Electronics and Nanotechnology,

- ELNANO 2018, Article No. 8477439 (2018). <https://doi.org/10.1109/ELNANO.2018.8477439>
5. Babichev, S., Sharko, O., Sharko, A., Mikhalyov, O.: Soft filtering of acoustic emission signals based on the complex use of Huang transform and wavelet analysis. *Adv. Intell. Syst. Comput.* **1020**, 3–19 (2020). https://doi.org/10.1007/978-3-030-26474-1_1
 6. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopolymers Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
 7. Bolstad, B., Irizarry, R., Astrand, M., Speed, T.: A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19**(2), 185–193 (2003). <https://doi.org/10.1093/bioinformatics/19.2.185>
 8. Das, A., Bhuyan, P.: Self-organizing tree algorithm (SOTA) clustering for defining level of service (LoS) criteria of urban streets. *Period. Polytech. Transp. Eng.* **47**(4), 309–317 (2019). <https://doi.org/10.3311/PPtr.9911>
 9. Dussiau, C., Boussaroque, A., Gaillard, M., et al.: Hematopoietic differentiation is characterized by a transient peak of entropy at a single-cell level. *BMC Biol.* **20**(1), 60 (2022). <https://doi.org/10.1186/s12915-022-01264-9>
 10. Ezugwu, A., Ikotun, A., Oyelade, O., et al.: A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng. Appl. Artif. Intell.* **110**, 104743 (2022). <https://doi.org/10.1016/j.engappai.2022.104743>
 11. Ganjalizadeh, V., Meena, G., Wall, T., et al.: Fast custom wavelet analysis technique for single molecule detection and identification. *Nature Communications* **13**(1), art. no. 1035 (2022). <https://doi.org/10.1038/s41467-022-28703-z5>
 12. Golub, T., Slonim, D., Tamayo, P., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**(5439), 531–537 (1999). <https://doi.org/10.1126/science.286.5439.531>
 13. Gomez, S., Rodriguez, J., Rodriguez, F., Juez, F.: Analysis of the temporal structure evolution of physical systems with the self-organising tree algorithm (SOTA): application for validating neural network systems on adaptive optics data before on-sky implementation. *Entropy* **19**(3), 103 (2017). <https://doi.org/10.3390/e19030103>
 14. Hayes, L., Basta, N., Muirhead, C., et al.: Temporal clustering of neuroblastic tumours in children and young adults from Ontario, Canada. *Environ. Health Glob. Access Sci. Source* **21**(1), 30 (2022). <https://doi.org/10.1186/s12940-022-00846-y>
 15. Liang, L., Li, J., Yu, J., et al.: Establishment and validation of a novel invasion-related gene signature for predicting the prognosis of ovarian cancer. *Cancer Cell Int.* **22**(1), 118 (2022). <https://doi.org/10.1186/s12935-022-02502-4>
 16. Lim, S., Lee, C., Tan, J., Lim, S., You, C.: Implementing self organising map to organise the unstructured data. *J. Phys. Conf. Ser.* **2129**(1), 012046 (2021). <https://doi.org/10.1088/1742-6596/2129/1/012046>
 17. Litvinenko, V.I., Burgher, J.A., Vyshemirskij, V.S., Sokolova, N.A.: Application of genetic algorithm for optimization gasoline fractions blending compounding. In: *Proceedings - 2002 IEEE International Conference on Artificial Intelligence Systems, ICAIS 2002*, pp. 391–394 (2002). <https://doi.org/10.1109/ICAIS.2002.1048134>

18. Liu, X., Zhao, J., Xue, L., et al.: A comparison of transcriptome analysis methods with reference genome. *BMC Genom.* **23**(1), 232 (2022). <https://doi.org/10.1186/s12864-022-08465-0>
19. Mohseni, M., Redies, C., Gast, V.: Approximate entropy in canonical and non-canonical fiction. *Entropy* **24**(2), 278 (2022). <https://doi.org/10.3390/e24020278>
20. Ramshaw, J.: Maximum entropy and constraints in composite systems. *Phys. Rev. E* **105**(2), 024138 (2022). <https://doi.org/10.1103/PhysRevE.105.024138>
21. Rosa, G.A., de Oliveira Ferreira, D., Pinheiro, A.P., Yamanaka, K.: Analysis of electricity customer clusters using self-organizing maps. In: Arai, K. (ed.) *IntelliSys 2021*. LNNS, vol. 295, pp. 312–325. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-82196-8_24
22. Sarswat, S., Aiswarya, R., Jose, J.: Shannon entropy of resonant scattered state in the e-c60elastic collision. *J. Phys. B Atom. Mol. Opt. Phys.* **55**(5), 055003 (2022). <https://doi.org/10.1088/1361-6455/ac5719>
23. Soni, N., Ganatra, A.: Categorization of several clustering algorithms from different perspective: a review. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2**(8), 63–68 (2012)
24. Whitaker, V., Oldham, M., Boyd, J., et al.: Clustering of health-related behaviours within children aged 11–16: a systematic review. *BMC Publ. Health* **21**(1), 21 (2021). <https://doi.org/10.1186/s12889-020-10140-6>
25. Xiong, K., Dong, Y., Zhao, S.: A clustering method with historical data to support large-scale consensus-reaching process in group decision-making. *Int. J. Comput. Intell. Syst.* **15**(1), 1–21 (2022). <https://doi.org/10.1007/s44196-022-00072-x>



Features of the Application of the Principal Component Method to the Study of Acoustic Emission Signals Under Loading of Multilayer Structures

Petr Louda¹ , Oleksandr Sharko² , Dmitry Stepanchikov³  ,
and Artem Sharko¹ 

¹ Technical University of Liberec, Liberec, Czech Republic
petr.louda@tul.cz

² Kherson State Maritime Academy, Kherson, Ukraine
sharkoartem@seznam.cz, avsharko@ukr.net

³ Kherson National Technical University, Kherson, Ukraine
dmitro_step75@ukr.net

Abstract. From the standpoint of fundamental consideration of experimental data on the loading kinetics of multilayer continua based on mechanical tests and acoustic emission measurements, large amounts of digital information on the parameters of AE signals during four-point bending of metal-based epoxy coatings have been processed and systematized. Features of the structure of the AE spectrum under load are reflected in the large size of the input data associated with the material's response to external force action and the generation of different AE signals from media. The principal components method was used to analyze the acoustic emission spectra under loading of multilayer structures. The relationship between the kinetics of structural changes at various stages of deformation of materials and the components of the acoustic spectrum is revealed and quantitatively described. Visualization of acoustic emission signals in the time domain reveals a tendency to increase their impulsiveness. The processing of the experimental data using the principal component method made it possible to cluster the AE signals. The correspondence of clusters with the stages of strain hardening is established. The results obtained can be used in the AE study of the stages of loading processes and diagnostics of multilayer structures. The main characteristics of the impulsivity of AE signals under loading of continuous conjugated media are calculated. Recommendations are given on the use of specific components of the principal component method as indicators of the state of strain hardening of materials of multilayer structures.

Keywords: Multilayer media · Mechanical testing · Acoustic emission · Loading · Principal component method

1 Introduction

Solving the problem of diagnosing the structural state of the material of multilayer continuums under load requires introducing new informative parameters and methods for their processing. One of the most effective methods for obtaining information about changes in the internal structure of materials under load is the acoustic emission (AE) method. The signals at the output of the AE diagnostic system under changing deformation conditions are characterized by various characteristics: amplitude, frequency, period, moments of occurrence, energy, density, intensity, and dispersion. They are discrete in time and quantitative in level i.e., digital. They are calculated by analog-to-digital conversions of the magnitudes of electrical signals from sensors, depending on the method receipt under loading. A large amount of information characterizes them. An analog-to-digital converter is the main key tool for obtaining information in AE diagnostics. The diagnostic procedure combines signals into clusters, classification, and analysis using the k -means method, principal component method, neural networks, and Kohonen maps. In this case, to eliminate interference, it is required to limit the input information by level and to quantize the AE signals in time to reduce the little informative values. Digital signals are obtained based on numerical methods and digital computer technologies.

The selection of uncorrelated combinations of features among correlated data allows the principal components method. The method is based on the idea of constructing a linear feature transformation, in which the signal-to-noise ratio in the new coordinate system decreases with increasing coordinate number. In this case, the data are presented as the sum of the useful signal and the uncorrelated noise.

When establishing the relationship between change in mechanical properties during the deformation of materials and the generation of appearance of AE signals, the change in spectral characteristics signals is taken as the basis [9, 36]. During loading of materials, AE is recorded only with the onset of microplastic deformation, i.e. AE is directly related to the movement of defects in the crystal lattice. This circumstance determined the loading method.

Four-point bending tests were chosen as the sources of AE signals under loading of multilayer structures. Among the numerous mechanical tests of the structure and properties of materials, four-point bending tests are considered the most time-consuming and complex, because. The sample experiences various deformations under loading. Its upper part is compressed, the lower part is stretched, and only the middle axial part, located between the loading indenters of the installation, is subjected to pure bending. For the case of coated materials, this situation is even more complicated due to the heterogeneity in structure and properties of the base and coating materials. This situation is of great practical interest due to the wide use of various protective coatings for products operating in aggressive environments, corrosion damage, mechanical abrasion during operation, temperature changes, humidity, etc. A special class of coatings is represented by coatings made of polymer composite materials on a metal basis - a new direction in improving the operational reliability of products.

Determining the degree of critical damage precedes the formation of cracks. This is not only an urgent scientific problem but also a technological and operational means of increasing the reliability of multilayer structures. Of great practical interest is studying energy processes in conjugated media, such as coatings on a metal base [30,31,43]. Thanks to the development of the principal component method, new possibilities appear for isolating weak signals even when their amplitude is significantly lower than the noise level. Identifying the state of conjugated structures is based on the isomorphism of the model and diagnostic system, the system of criteria and factors for their evaluation.

The aim of this research is to study the acoustic spectrum of continuous multilayer media during loading with the use of the principal component analysis method.

2 Literature Review

Digitization of the structure of AE spectrum of products under load is based on establishing a connection between the informational features of the AE and the characteristics of deformation mechanisms [10,11,15,23]. In [26,32,35], the results of digital diagnostics of deformation effects are described with the information stored in the memory of a storage device. In [39], the AE phenomenon revealed defects in composite materials. The detection of cracks and fatigue damage using the AE method is presented in [18,21,46,55]. In [22], the results AE diagnostics during the bending of steel sheets are presented in [3,40] for recognizing changes in the structure of materials and destruction. In [37,44,45], an information-entropy model of the quality of input information is presented. To exclude the subjective factor in determining the discrimination threshold that determines the quality of input information, the experience borrowed from related branches of information support for management under uncertainty is useful [5–7,28]. The most informative AE parameters under bending and uniaxial loading are presented in [29]. The technology for determining the residual resource of metal products under combined load conditions by AE measurements is introduced in [33]. In [38], identifying AE bursts coming from triaxial tests during the deformation of steel pipes with a protective coating is presented by classifying the physical-statistical parameters obtained by the principal component method. In [13], the results of applying the main component method to the study of AE signals with a change in the stress-strain state of aluminum alloys are described.

In [17], the mechanisms of damage to composites were assessed using the methods of multiparameter analysis based on AE data. The relationship between principal component analysis, k -means and Kohonen maps is presented. In [14], the damage mechanism to composites was studied using the wavelet packet transform of fuzzy clustering and principal component analysis. In [16,50], an integrated method for identifying fatigue damage is described by analyzing the main components of an artificial neural network and information entropy. In [19], AE signals were processed using time-frequency analysis to diagnose bearing failures.

The condition of the bearing material was determined by a multivariate analysis of the main AE components. In [27,48], the principal component method was used to reduce the size of AE data when drilling composite materials. In [4], the AE wave clustering method was used to monitor the state of structures during bending tests, in [54] for compression, in [25,34,41] to monitor the development of damage during static and dynamic tests, in [52] for torsion tests three-dimensional braided composite rolls. The AE signals were analyzed using principal components and fuzzy clustering methods.

In [49], the AE and digital image correlation method were used to identify damage to composite laminates using the principal component method, in [47] to study high-pressure vessels made of maraging steel for aerospace applications, and in [20] to monitor the state of wind turbines. In [53], a k -means clustering algorithm was proposed in combination with principal component analysis, which was used to determine the tensile strain of woven composites and digital image correlation.

Cluster analysis of AE signals in combination with infrared thermography was used in [51] to analyze the evolution of defects in glass-epoxy composites. When analyzing AE signals, the k -means and principal component methods were used. It is shown that the AE signals during tensile loading can be divided into four clusters of damage types such as matrix cracking, fiber detachment from the substrate, delamination, and fiber detachment. In [8], clustering of damage mechanism features was used to process AE data in the time and frequency domains. In [24], the technology of neural networks with self-organizing display of signs of the distribution of the characteristic parameters of AE was used to assess corrosion damage on prestressed steel filaments. Cluster analysis of AE signals made it possible to identify four types of damage sources and evaluate their evolution.

The principal component method makes it possible to visually represent data and simplify the calculation and interpretation of compressed volumes of input information. The analysis of the features of the study of acoustic emission signals under loading of multilayer structures will allow us to discover new effects that occur when they are loaded.

3 Methodology

The principal component analysis method approximates an n -dimensional cloud of observations to an n -dimensional ellipsoid whose semi-axes will be the main components. In the method of principal components, there are subspaces. The projection on which the root-mean-square distance between points is maximal, i.e. the resulting distortion resulting from projections is minimal. Using the principal components method is reduced to the approximation of data by linear manifolds of lower dimensions. The primary component method is used in statistical measurement problems in which the signal to be measured is not directly observed.

The essence of the method is to use the decomposition of signals in special bases formed from the eigenvectors of the information determinant. This method

is an orthogonal linear transformation that maps data from the original feature space to a new lower-dimensional space. In this case, the new coordinate system's first axis is constructed so that the dispersion of data along it is maximum. The second axis is orthogonal to the first so that the dispersion along it would be the maximum of the remaining possible ones, and so on. The first axis is called the first principal component. The second is called the second, and so on. When projected onto these axes, the greatest amount of information is stored.

The principal component analysis method is applied to data written as a matrix of numbers X . The input data matrix has the size $n \times m$, where n is the number of observation objects, m is the the number of their analytical features. If we have two features that are strongly correlated with each other, then using the method of principal components, you can find their combinations with the loss of part of the information, expressing both features with one new one. To prepare the data, the sample should be centered by shifting it so that the average value of the features is equal to zero. In this case, the input data matrix X is transformed into a normalized centered values Y matrix. The principal component method begins with solving the problem of the best approximation of a finite set of points by straight lines and planes. To do this, for a finite set of vectors $x_1, x_2, \dots, x_m \in R^n$ need to find the value $S_k \in R^n$, for which the sum of squared deviations x_i from S_k will be minimal. Here $k = 0, 1, 2, \dots, n - 1$ linear manifold in R^n .

$$\sum_{i=1}^m \text{dist}^2(x_i, S_k) \longrightarrow \min, \tag{1}$$

where $\text{dist}(x_i, S_k)$ is the Euclidean distance from a point to a line.

The solution of the approximation problem for k is given by a set of nested manifolds S . These manifolds, in turn, are determined by an orthonormal set of principal component vectors. These manifolds are in turn determined by an orthonormal set of vectors $\{a_1, \dots, a_{k-1}\}$ main components and vector a_0 , which is determined by minimizing S_0 .

$$a_0 = \arg \left(\sum_{i=1}^m \text{dist}^2(x_i, S_k) \right) = \underset{a_0 \in R^n}{\text{argmin}} \left(\sum_{i=1}^m |x_i - a_0|^2 \right), \tag{2}$$

The variational definition of the mean as the point that minimizes the sum of squared distances is:

$$a_0 = \frac{1}{m} \sum_{i=1}^m x_i^2. \tag{3}$$

When projected onto these axes, the greatest amount of information is stored. The first principal component maximizes the sample variance of the data projection. It is necessary to find such an orthogonal transformation to a new coordinate system for which the sample variance of the data along the first coordinate is maximum.

The second principal component, provided that the first coordinate is orthogonal, maximizes the sample variance of the data along the second coordinate.

The k -th principal component, under condition of orthogonality of the $k - 1$ coordinate, maximizes the sample variance of the data along the values of the $k - 1$ coordinate. Solving the best approximation problem gives the same set of principal components as the search for orthogonal projections with the greatest scattering. The problems of determining the principal components are reduced in their methodological plan to the problem of diagonalizing the sample of the covariance matrix.

The operations of the methodology for using the method of principal components and finding specific components are the following:

- centering the data by subtracting the mean $x_i := x_i - \overline{X}_i$, where symbol $:=$ means equality by definition;
- calculation of the first principal component by solving the problem:

$$a_1 = \operatorname{argmin}_{|a_1|=1} \left(\sum_{i=1}^m |x_i - a_1(a_1, x_i)|^2 \right); \tag{4}$$

- subtraction from projection data to the first principal component:

$$x_i := x_i - a_1(a_1, x_i); \tag{5}$$

- finding the second principal component as a solution to the problem:

$$a_2 = \operatorname{argmin}_{|a_2|=1} \left(\sum_{i=1}^m |x_i - a_2(a_2, x_i)|^2 \right); \tag{6}$$

- projection subtraction on $k - 1$ principal component:

$$x_i := x_i - a_{k-1}(a_{k-1}, x_i); \tag{7}$$

- finding the k -th principal component:

$$a_k = \operatorname{argmin}_{|a_k|=1} \left(\sum_{i=1}^m |x_i - a_k(a_k, x_i)|^2 \right). \tag{8}$$

Received vectors $\{a_1, \dots, a_{k-1}\}$ are orthonormal. Thus, the methodology for finding the principal components is that at each preparatory step $(2k - 1)$ subtract the projection onto the previous principal component.

To describe a multidimensional random variable, in addition to the mathematical expectation E and the dispersion of its projections on the axis, one should use the concept of the covariance matrix of pair correlations $R(S)$, those a matrix whose elements are feature correlations x_i, x_j . The inter-feature covariance matrix describes the combined effect of several variables. The covariance formula is

$$\operatorname{Cov}(x_i, x_j) = E[(x_i - E(x_i)) \cdot (x_j - E(x_j))] = E(x_i, x_j) - E(x_i) \cdot E(x_j). \tag{9}$$

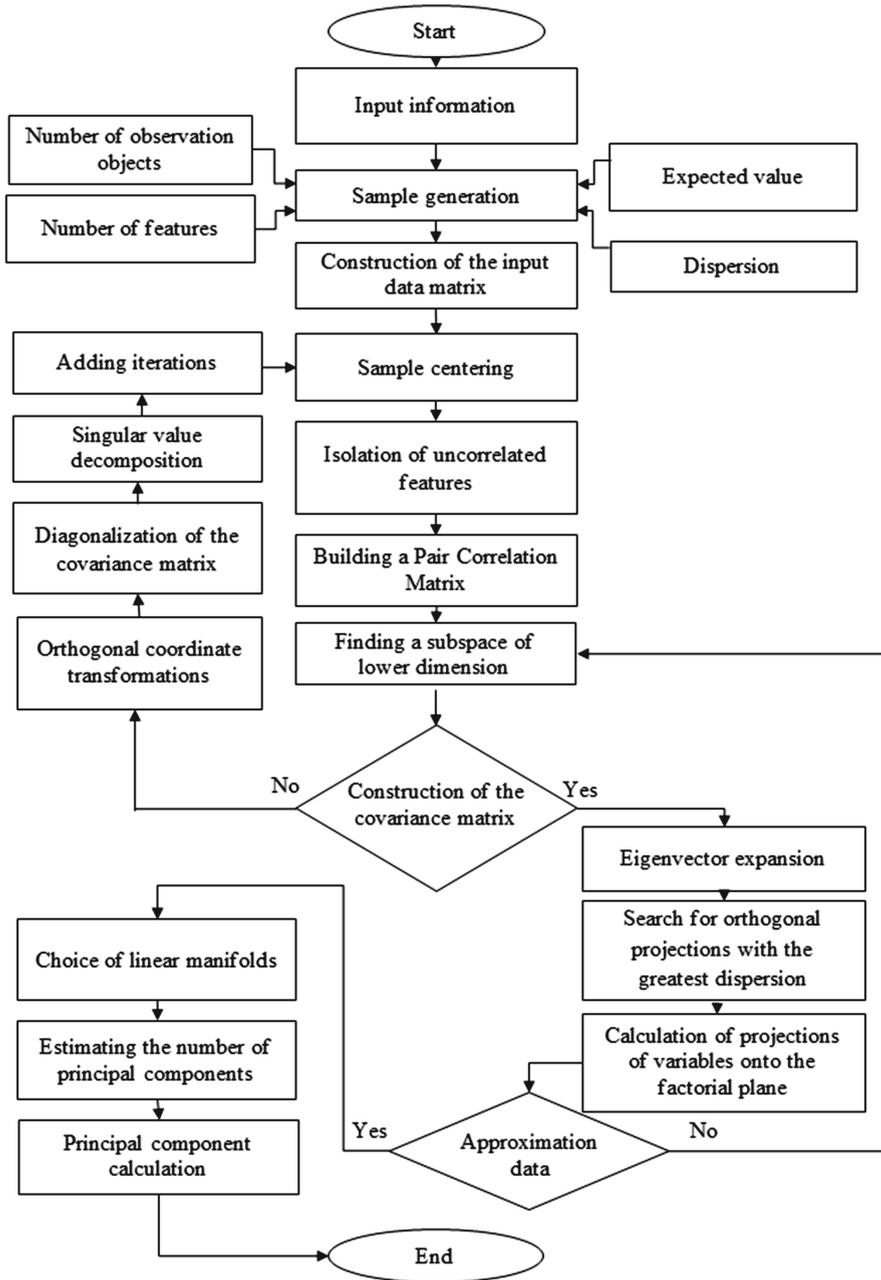


Fig. 1. Algorithm of iterative procedures for constructing principal components

In such a covariance matrix, there will be variances of features x_i, x_j along the diagonal, and in the remaining cells there will be covariances of the corresponding pairs of features. The covariance matrix can be represented as eigenvectors and eigenvalues. The eigenvalues are used to estimate the contribution of the principal components to the overall variability. The dispersion along the eigenvectors is proportional to the eigenvalues. The eigenvectors define the directions of the principal components. With their help, you can find new axes in the feature space, along which there will be a maximum separation of points. The calculation of the principal components is reduced to the calculation of the singular value decomposition of the data matrix and eigenvectors and the eigenvalues covariance matrix original data. Singular value decomposition is a decomposition of a real matrix, which shows the geometric structure of the matrix and allows you to visualize the available data.

The mathematical content of the principal component method assumes a predetermined accuracy in the presence of a useful signal, relatively small dimension and large amplitude, and noise of large dimension and small amplitude. The useful signal is contained in the projection on the first principal components, then the fraction of noise in the remaining components is much higher.

As a result of the study, it is assumed that the orthogonal components in the spectrum AE signal will be separated, followed by removing the noise component to increase the informative part of the signal.

The compiled algorithm of iterative procedures is shown in Fig. 1.

4 Materials and Methods

Modeling the processes of product operation under conditions of uncertainty of the influence of loads is directly related to its information support. An effective method for studying the dynamic processes of changing the structure of materials during the operation of metal structures and identifying their parameters is the method of active experiment.

Progress in the use of new composite materials and multilayer structures based on them is determined by the diagnosis of their changes under loading. This serves as the basis for choosing a material for research, which was chosen as multilayer conjugated media coated with polymeric materials on a metal basis.

As research materials, we used the results of synchronous AE measurements and mechanical tests for four-point bending, performed for a coating of epoxy resin ED-20 on a metal base made of St3sp steel. The choice of epoxy resin as research material is explained by the fact that it is characterized by high adhesion to metal, glass and concrete. The choice of material for the substrate was explained by the wide distribution of this steel grade in various technical applications. The physical and mechanical properties of resin ED-20 are presented in Table 1. The mechanical properties of St3sp steel and its international counterparts are shown in Tables 2 and 3.

Table 1. Physical and mechanical properties of epoxy resin ED-20

Tensile stress, MPa	50
Compressive stress, MPa	113
Breaking stress in bending, MPa	115
Modulus of elasticity, N/mm ²	3000–4500
Tensile strength, N/mm ²	80

Table 2. Mechanical properties of steel of St3sp grade

Standard	σ_B , MPa	$\sigma_{0.2}$, MPa	δ , %
ISO 4995–78	370–490	206–245	23–26

Table 3. International analogues steel of St3sp grade

Germany	USA	Japan	France	Belgium	China
1.0038	A284GrD	SS330	E24–2	FE360BFN	Q235A
St37–3	M1017	SS400	S234JRG2	FED1FF	Q235B

As a method for studying AE signals under loading of multilayer structures, the method of principal components was used. The principal component analysis is based on logical assumptions that the values of a set of interrelated features generate a new result.

5 Experiment, Results and Discussion

One of the complex deformation effects on the material during its loading is bending deformation. During bending, an inhomogeneous plane biaxial stress state occurs in the sample. As a result, the transverse deformation becomes more complicated: the lower part of the sample is stretched while the upper part is compressed. The stresses that are associated with the magnitude of the bending moment are different in magnitude in the section of the sample. The maximum stresses are created near the sample surface. This feature complicates the assessment of true stresses and strains. In this work, the goal was to reveal the features of plastic deformation and the possibilities of fixing the occurrence of AE signals during the loading of multilayer structures.

Identification of the structural features of the deformation mechanisms was carried out when the samples were deformed to a four-point bend. The scheme of testing for four-point bending is shown in Fig. 2.

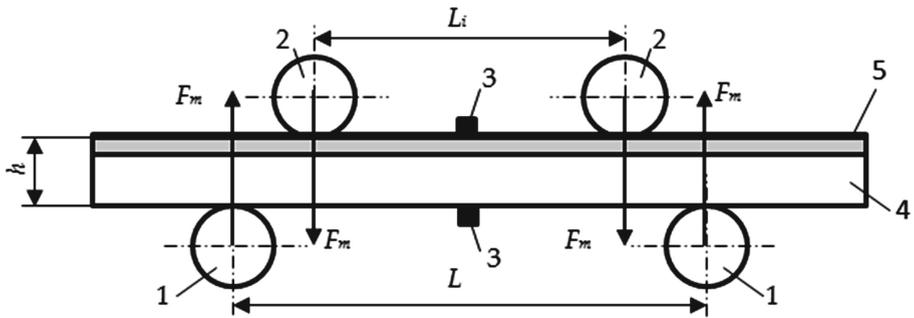


Fig. 2. Four-point bending test scheme: 1 – lower support, 2 – upper support, 3 – AE sensors, 4 – metal substrate, 5 – epoxy coating

The formula calculated mechanical stress during four-point bending

$$\sigma = \frac{3F_m(L - L_i)}{2bh^2}. \tag{10}$$

where: F_m is a load, N; L is the distance between the lower supports, mm; L_i is the distance between upper supports, mm; b is the sample width, mm; h is the average sample thickness, mm.

During testing, the sample was installed between the prisms testing machine and the identifier loading device. A counter for reporting time from the start of loading was installed on the drive device. To assess the degree of loading of the sample, simultaneously with the timer, the signal was sent to the load cell and to the digital deflection indicator, and then through the microprocessor to the computer. The signal from the AE sensor was also fed here. The recording device was made in the form of a brand storage oscilloscope (RIGOL DS 1052E Digital oscilloscope). Thus, information was simultaneously collected on the computer from the parameters of the loading device, the deflection, and the moments of occurrence of AE signals. Tests were carried out in compliance with ASNT, ASTM, ISO9001 [1,2] standards. AE sensors were installed both on the side of the metal base and on the side of the composite coating. During the experiment, 897 AE signals were registered. Figure 3 shows signals that exceeded the minimum threshold level of 0.01 V.

The location of the signals corresponds to the order in which they occur under loading. The peculiarity and originality of such representation of AE signals in Fig. 3 lies in the desire to display not so much the dynamics sequence of the loading process fixed by the force-measuring device but the synchronous display of the moments of visualization of AE bursts by the deflection indicator.

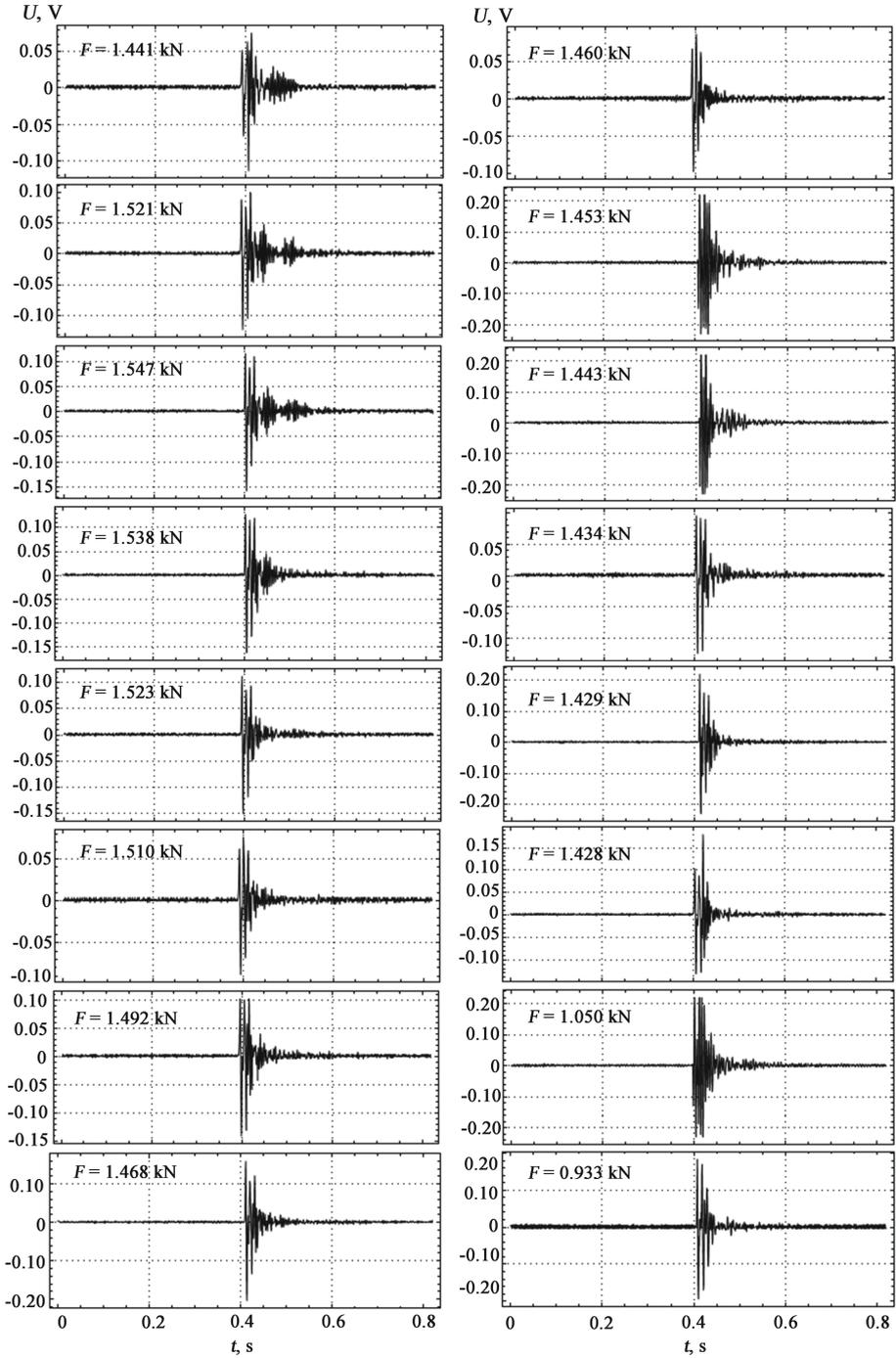


Fig. 3. Visualization of measurement results of coated samples under different loads

It can be seen from a consideration of Fig. 3 that the signals at the initial stage of loading are small, and a sharp increase in activity falls on the yield strength region. In the spectrum of AE signals in the stresses above the yield point, low-amplitude signals predominate.

Figure 4 shows on a single scale the entire set of AE signals obtained in the order of their occurrence and fixation during sample loading, which gives a visual representation of the change in peak amplitudes. Table 7 presents the experimental values of the main characteristics for four-point bending of the test sample.

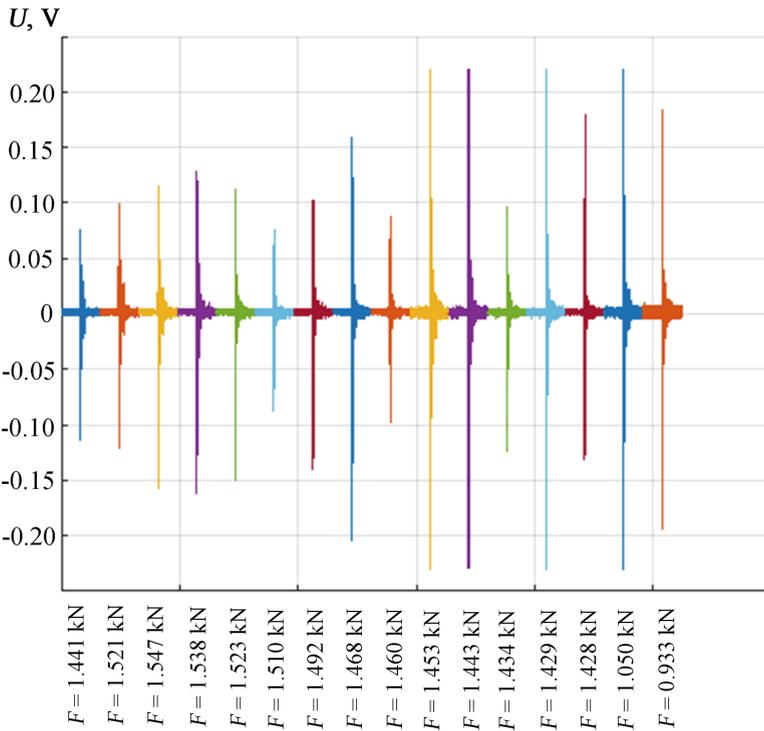


Fig. 4. The sequence of received AE signals in the order of their occurrence and fixation during sample loading

Studies that influence mechanical properties and the profile of the coating-substrate composition interface on the development of bending deformation showed that due to the incompatibility deformation of the base material and the coating, mesostress concentrators arise at the interface, which generates cracks in the coating and mesobands in the substrate. The quantitative and qualitative configuration zones of localized plastic flow and characteristics of the loading diagram for the coated specimens differ from those of the St3sp steel used as the

Table 4. Experimental values of the main characteristics for four-point bending at the moments of successive fixation of AE signals

No. AE measurements	Loading force F , kN	Mechanical stress σ , MPa	Sample deflection ΔL , mm	AE signal amplitude U_{max} , V
1	1.441	353.367	2.346	0.114
2	1.521	372.737	2.753	0.122
3	1.547	379.201	3.052	0.158
4	1.538	376.967	3.565	0.162
5	1.523	373.362	4.920	0.150
6	1.510	370.238	5.144	0.088
7	1.492	365.671	5.246	0.140
8	1.468	359.904	5.423	0.204
9	1.460	357.837	5.442	0.098
10	1.453	356.203	5.520	0.230
11	1.443	353.776	5.550	0.230
12	1.434	351.541	5.578	0.124
13	1.429	350.363	5.595	0.230
14	1.428	350.147	5.596	0.180
15	1.050	257.527	8.103	0.231
16	0.933	228.833	9.047	0.194

base. This difference consists absence of a yield point metal base and the presence of oscillations associated with the appearance of microcracks in the coating. The yield strength of coated specimens is significantly lower than that of uncoated specimens. The loading diagram for the layer is characterized by even lower indicators of the initiation of plastic flow zones. The plastic zone is formed in the form of bands that spontaneously develop across the sample, then the trajectories and directions change. The plastic deformation mechanisms can be considered depending on the scale levels on the basis of mesoconcepts about the dislocation structure of materials [30]. Plastic deformation in continuum mechanics considers the translational motion of defects under the action of applied stress. The plastic flow curve is obtained by calculating the strain hardening above the yield strength of the material. The hierarchy of structural levels of plastic deformation of solids can vary from three stages, including elastic deformation, elastic-plastic and fracture, to five or more, depending on the material's structure (Table 4).

Plastic flows develop as a sequential process of loss of shear stability at different scale levels: micro, meso and macro. At the micro level, the loss of shear stability occurs in local zones of the crystal lattice, leading to dislocations. At the mesolevel, mesobands of localized deformation are formed. Their formation and development leads to stress relaxation. At the macro level, the transition

of a material from an elastic state to a plastic state is based on physical force interactions. The structural levels of deformation of solids have become a new paradigm that reveals the possibilities and patterns of behavior of deformable conjugated continuous media with metal-based coatings.

Stages characterize the stress-strain curve for conjugated media with different slopes of the strain coefficient $K = d\sigma/dL$ and the intensity of deformations. The duration of the stages depends on the material, structure, and type of loading.

The main regularities of plastic flow under load and the hierarchy of structural levels are associated with the formation of dissipative mesostructures and fragmentation of the deformed material. Destruction is the final stage of fragmentation. The mechanism of structure degradation in the early stages of the deformation process is phenomenologically based on the localization of macroscale levels. The stress-strain curve reflects the strain hardening of the material above the yield strength and the relaxation processes associated with local loss of shear stability.

Table 5. Statistical characteristics of AE signals in the time domain

No. AE signal	Mean	Std	Skewness	Kurtosis	Peak2Peak	RMS	CrestFactor	ShapeFactor	ImpulseFactor	MarginFactor	Energy
1	0.0015	0.0093	-1.5788	46.1495	0.1900	0.0094	8.0987	2.4808	20.0917	5311.5	0.1442
2	0.0013	0.0114	-0.8780	44.8153	0.2220	0.0115	8.7164	2.7725	24.1664	5840.2	0.2156
3	0.0014	0.0130	-1.1834	52.0257	0.2740	0.0131	8.8484	2.8077	24.8441	5321.0	0.2815
4	0.0013	0.0141	-1.3705	58.8446	0.2900	0.0142	9.0409	3.2058	28.9831	6562.7	0.3283
5	0.0014	0.0115	-1.7775	68.1377	0.2620	0.0116	9.6473	3.0851	29.7625	7909.0	0.2208
6	0.0012	0.0081	-0.2334	51.3737	0.1640	0.0081	9.3327	2.8070	26.1970	9030.0	0.1086
7	0.0014	0.0123	-1.7247	59.1929	0.2420	0.0124	8.2321	3.0705	25.2762	6263.6	0.2515
8	0.0013	0.0149	-2.0050	76.7052	0.3640	0.0149	10.7148	3.4823	37.3121	8701.2	0.3652
9	0.0013	0.0087	-0.0159	56.9910	0.1860	0.0087	10.0646	2.8945	29.1318	9643.9	0.1252
10	0.0011	0.0282	-0.0140	39.5461	0.4500	0.0282	7.8033	3.3133	25.8545	3038.4	1.3020
11	0.0009	0.0254	-0.5172	53.1266	0.4500	0.0254	8.6724	3.5346	30.6533	4271.0	1.0541
12	0.0013	0.0111	-1.6224	61.4874	0.2200	0.0111	8.6120	3.0895	26.6071	7374.4	0.2035
13	0.0014	0.0186	-1.1460	74.4964	0.4500	0.0187	11.7807	3.6450	42.9409	8381.5	0.5712
14	0.0013	0.0139	0.7712	66.8411	0.3120	0.0139	12.9304	3.4580	44.7134	1110.7	0.3174
15	0.0012	0.0282	0.1797	40.9413	0.4500	0.0282	7.8087	3.3119	25.8619	3040.2	1.3002
16	0.0014	0.0174	-1.2202	63.2011	0.3780	0.0175	10.5384	2.8221	29.7407	4807.1	0.4993

Stage I is characterized by the formation of structural defects during bending in the near-surface layer. Stage II is associated with sample fluidity and is characterized by a high intensity of AE signals. Stage III is characterized by the integral accumulation of defects and the formation of microcracks. Stage IV reflects the development of microcracks and the formation of larger cracks. An intense plastic flow occurs with the involvement of new volumes of material in the deformation process. Stage V is the pre-fracture stage and is a development of stage IV. The boundaries between these stages are accompanied by the rapid growth of one of the cracks and the destruction of the sample. Signals with higher energy appear and characterize the growth of main macrocracks.

To determine the boundaries of the stages of plastic deformation, the values of the strain hardening coefficient were calculated from the measured values of

the load and deflection. From the inflections of the dependencies of the values of this coefficient on strain, it is possible to more correctly determine the boundaries of the stages than directly from the stress-strain curve [12].

For the analysis and processing of AE signals, the statistical characteristics of AE signals in the time domain were chosen, such as the mean value (Mean), standard deviation (Std), skewness (Skewness), kurtosis (Kurtosis), full swing (Peak2Peak), root mean square value (RMS), crest factor (CrestFactor), form factor (ShapeFactor), impulse factor (ImpulseFactor), marginal factor (MarginFactor), energy (Energy), as well as statistical characteristics of AE signals in the frequency domain, such as the average value (SKMean), standard deviation (SKStd), skewness (SKSkewness), kurtosis (SKKurtosis).

The values of these characteristics for experimental AE signals are given in digital form in Tables 5 and 6.

Table 6. Statistical characteristics of AE signals in the frequency domain

No. AE signal	Mean	Std	Skewness	Energy
1	3.4311	4.5471	1.0117	2.6877
2	4.8150	3.6392	0.5833	3.0841
3	5.5862	4.8046	0.5220	2.0713
4	8.0349	4.7959	-0.0636	1.9321
5	4.7592	4.7597	0.4639	1.6341
6	4.2540	4.3964	0.4511	1.6952
7	6.1587	4.3130	0.2593	1.9446
8	6.6867	5.3275	0.4322	1.9900
9	4.6347	4.5982	0.4262	1.7772
10	8.0498	4.2910	-0.0834	2.5744
11	9.7963	4.8539	-0.3245	2.1791
12	4.4324	5.3848	0.8855	2.3181
13	9.7147	4.4124	-0.4406	2.6192
14	5.8557	6.1746	0.4560	1.5558
15	8.4443	4.5069	-0.3105	2.0073
16	10.2873	4.9959	-0.7369	2.4156

The main criterion for the belonging of the statistical characteristics of AE signals under loading of multilayer structures in the frequency and time domains to their further analysis is the numerical value of monotonicity. In [42], for a

quantitative assessment of the belonging of statistical characteristics to their further analysis, it was proposed to use the formula

$$\begin{aligned} \text{Monotonicity}(x_i) &= \\ &= \frac{1}{m} \sum_{j=1}^m \left| \frac{\text{Number of positive diff}(x_i^j) - \text{Number of negative diff}(x_i^j)}{n-1} \right|, \end{aligned} \quad (11)$$

where: n is the number of measurement points, in our case $n = 16$. m is the number of controlled samples, in our case $m = 1$, x_i^j is i -th characteristic measured on the j -th sample, $\text{diff}(x_i^j) = x_i^j(t) - x_i^j(t-1)$.

The results of calculating the monotonicity of the statistical characteristics of AE signals in the frequency and time domains according to formula (11) are shown in Fig. 5.

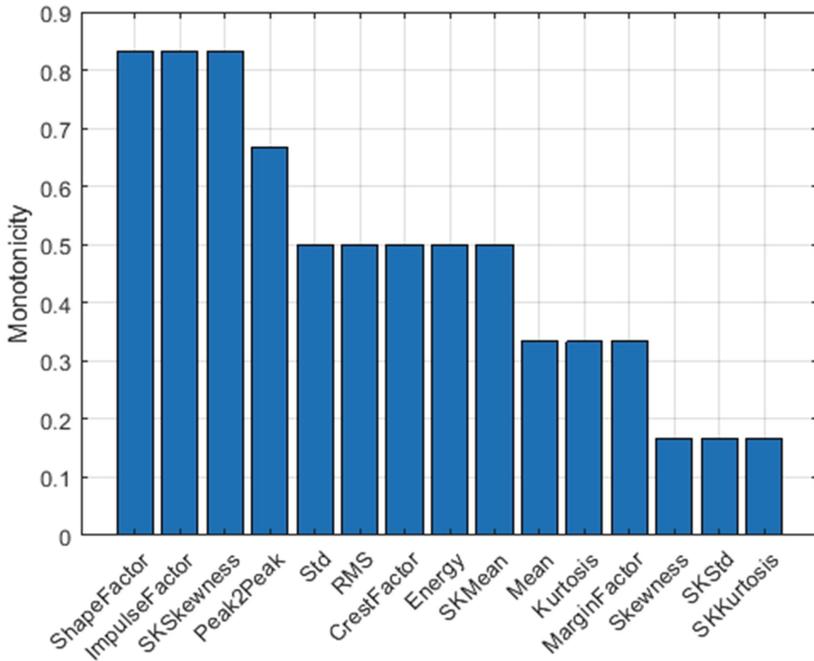


Fig. 5. Monotonicity of statistical characteristics of AE signals arising in the process of sample loading

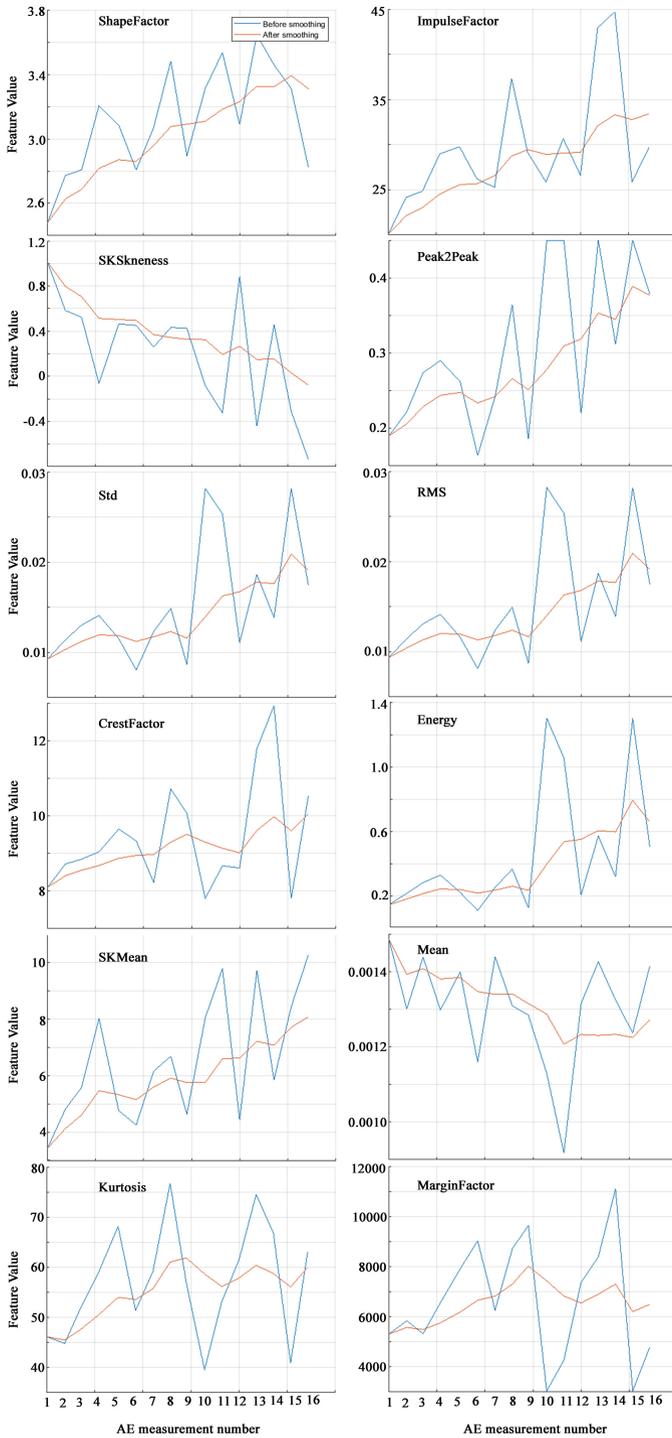


Fig. 6. Comparison of noisy (blue line) and filtered (red line) characteristics of AE signals with monotonicity ≥ 0.3

Characteristics with a monotonicity score of more than 0.3 are selected for their combination in subsequent analysis using the method of principal components.

The obtained statistical characteristics are associated with noise. Causal moving average filtering with a lag window of 5 steps must be applied to the extracted features. The term "causal" lag windowed moving average means that no future value is used in the filtering.

The results of comparing the noisy and filtered characteristics of AE signals with a monotonicity of ≥ 0.3 are shown in Fig. 6.

To reduce the dimensionality of the analyzed quantities and combine the signs of AE signals under loading of multilayer structures, the principal component analysis (PCA) was used in the work. Using the Matlab 2018b computer mathematics system, when processing the data in Tables 5 and 6, the first four principal components of PCA1, PCA2, PCA3 and PCA4 were calculated. Their distribution depending on the number of measurements, which correlates with the successive increase in deflection sample during tests for four-point bending, is shown in Fig. 7.

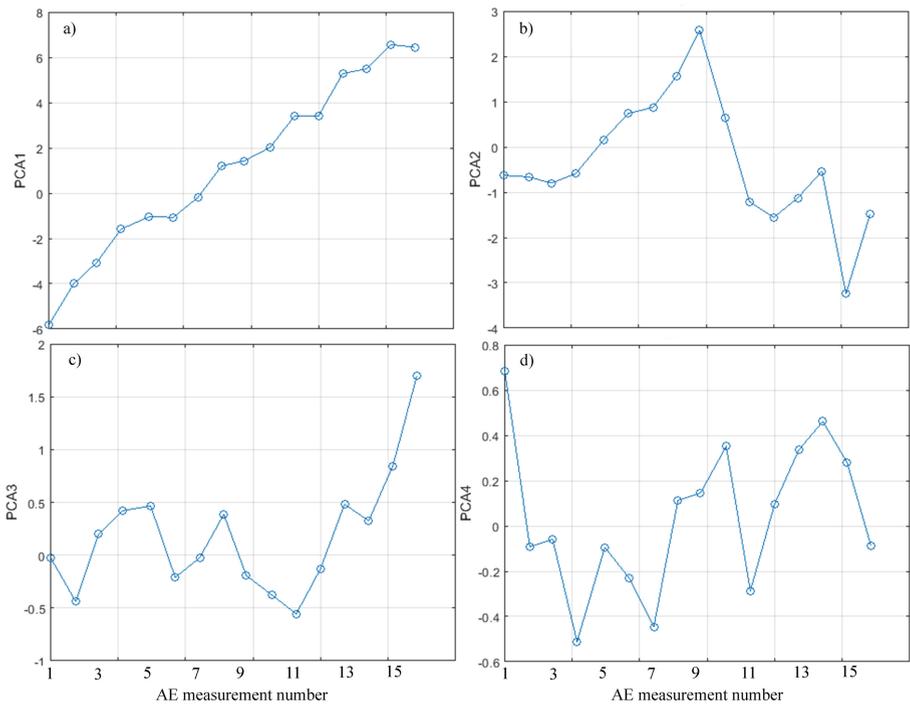


Fig. 7. First (a), second (b), third (c), and fourth (d) principal components for successively recorded AE signals in bending tests

Table 7. Values of the first four principal components of PCA1, PCA2, PCA3, and PCA4 for four-point bending of specimens at the instants of successive fixation of AE signals

No. AE measurements	Mechanical stress σ , MPa	Sample deflection ΔL , mm	PCA1	PCA2	PCA3	PCA4
1	353.367	2.346	-5.810	-0.625	-0.026	0.681
2	372.737	2.753	-3.997	-0.661	-0.438	-0.092
3	379.201	3.052	-3.062	-0.797	0.200	-0.058
4	376.967	3.565	-1.583	-0.580	0.422	-0.512
5	373.362	4.920	-1.052	0.162	0.465	-0.094
6	370.238	5.144	-1.064	0.742	-0.210	-0.231
7	365.671	5.246	-0.187	0.869	-0.023	-0.445
8	359.904	5.423	1.203	1.570	0.385	0.112
9	357.837	5.442	1.426	2.575	-0.192	0.145
10	356.203	5.520	2.031	0.635	-0.377	0.351
11	353.776	5.550	3.402	-1.206	-0.555	-0.287
12	351.541	5.578	3.415	-1.558	-0.130	0.095
13	350.363	5.595	5.278	-1.125	0.482	0.334
14	350.147	5.596	5.491	-0.536	0.325	0.461
15	257.527	8.103	6.557	-3.235	0.843	0.281
16	228.833	9.047	6.439	-1.483	1.699	-0.087

The resistance to plastic deformation is determined from the deformation diagram in coordinates σ , ΔL . The stress required to create plastic deformation determines the yield strength. Figure 8 shows the loading diagram of the sample under study made of St3sp steel with an epoxy coating based on ED-20 resin. The strain hardening intensity is expressed in terms of the strain hardening coefficient $K = d\sigma/dL$. The results of constructing the deformation curve according to the data force-measuring device when changing deflection ΔL as a dependency $K = f(\Delta L)$ are also presented in Fig. 8. Considering the above graphs, five successively alternating stages of strain hardening can be distinguished, at each of which energy is accumulated by the structure with the emission of AE pulses due to translational shifts.

The analysis of the stages presented in Fig. 8 AE allows not only to reduce the time and laboriousness of evaluating structural transformations during bending, but also to obtain real-time information about the processes occurring at all levels of damage accumulation. To obtain the necessary information about the onset and course of the stages of strain hardening with the specification boundaries and the sequence onset of the stages, the method of principal components allows.

Figure 9 shows the nature of the change in the principal components of PCA1, PCA2, PCA3 and PCA4 depending on the deflection of the sample. A spline

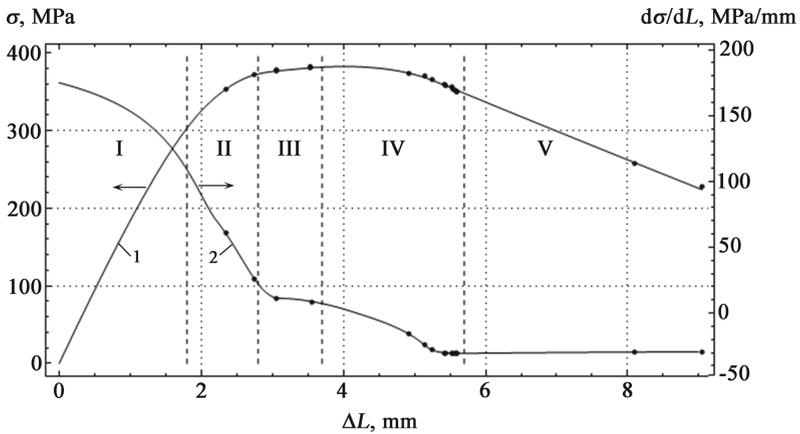


Fig. 8. Dependence of mechanical stress (curve 1) and strain hardening coefficient (curve 2) on deflection at four-point bending. Points on the graph show the moments of occurrence of AE signals, I-V are the stages of strain hardening

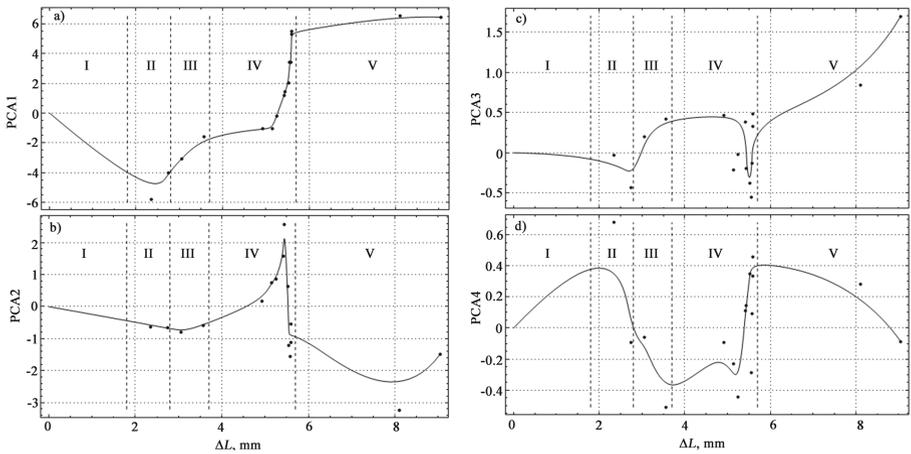


Fig. 9. The first (a), second (b), third (c) and fourth (d) principal components as functions of the sample deflection (the dots show the values of the principal components for recorded AE signals, the solid line is the cubic spline approximation)

approximation is used for a more visual display of the change in the main components, shown on the graphs by a solid line.

Comparison of Figs. 8 and 9 reveals the identity of the hardening stages, and if in Fig. 8 the division into stages was of a qualitative nature, then in Fig. 9 their quantitative boundaries are indicated, obtained on the basis of experimental data processing.

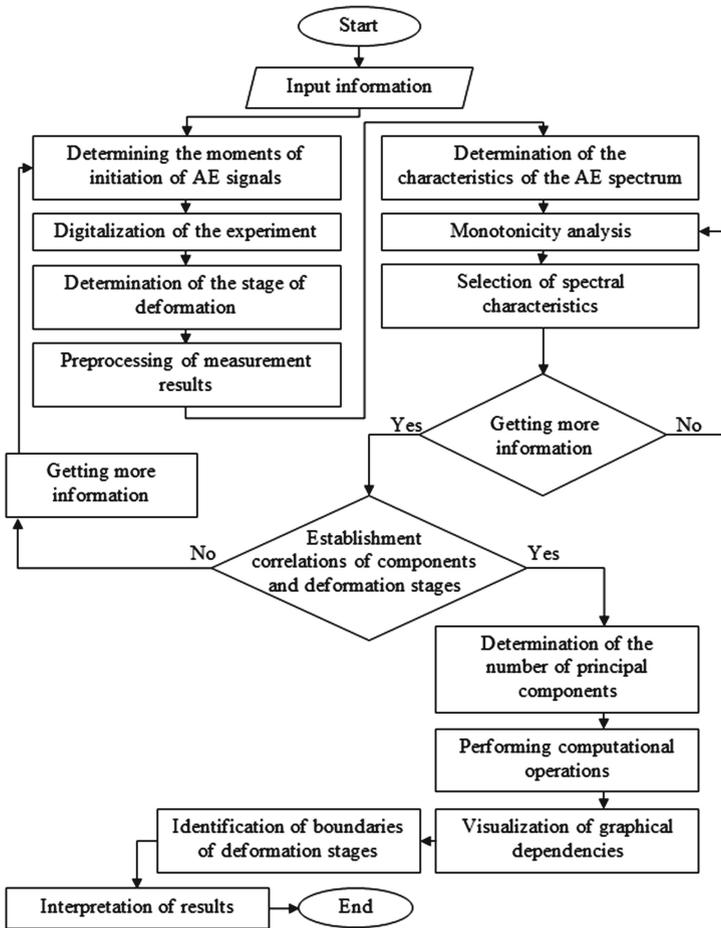


Fig. 10. Algorithm for implementing the principal component method in problems of diagnosing the state of materials under loading

The presented results of calculations of the features of the application of the principal component method to the study of acoustic emission signals under loading of multilayer structures make it possible to extract the values of information about structurally sensitive factors for further use with other methods of analyzing the state of multilayer conjugated continua under loading. Suppose there is no reason to expect that any semantic hierarchy is reflected in the variety of strength and physicomaterial properties of base and coating materials of different modulus. In that case, one should use the ordination of features, which consists in their ordering and arrangement in the space of the main components.

Comparison of Figs. 7 and 9 makes it possible to identify a number of characteristic features that can be used as the basis for practical recommendations for assessing the stages of deformation of materials under loading. Over the entire

load range of the stress-strain curve, good results are obtained using the first component, which changes under loading reveal a monotonous growth. Additional information for estimating the stages of deformation using the method of principal components is characterized by the optimal use of the fourth component for estimating the second stage of plastic deformation, also the fourth component - for estimating the third stage of plastic deformation, the second component - for estimating the fourth stage, the third component - for estimating the fifth stage. Recommended components in their influence can overlap and complement each other.

The algorithm for implementing the method of principal components in the problems of diagnosing the state of materials under loading is shown in Fig. 10. The use of the principal component method clarifies and explains a number of new effects of the staged hardening of multilayer structures in bending.

6 Conclusions

A feature of the application of the principal component method to the study of acoustic emission signals under loading of multilayer structures manifested itself in the presence of five stages of plastic deformation. Principal component analysis allows one to reduce the dimension of the data and obtain several principal components instead of the original features for ease of interpretation.

The main components used in the work do not depend on each other. Principal components can be treated as complex features and relationships between objects can be described in terms of these features. Principal components make it possible to unambiguously determine the share of use of each specific component in the overall variability of the loading process of multilayer structures.

The use of the principal component method makes it possible to explain a number of new effects in the bending of multilayer structures. The monotonic increase in the first principal component with increasing deflection is the essence of the recommendation for assessing the stages of plastic deformation over the entire range of the loading curve. To assess the stage of the onset of flow and high intensity of AE signals, it is recommended to use the fourth component, to assess the stage associated with the formation of microcracks, the fourth component is also recommended. To assess the stage of development of microcracks and the formation of macrocracks, it is recommended to use the second component; for the pre-fracture stage, it is recommended to use the third component. The considered principle of scale invariance can serve to analyze the complex nature of plastic deformation and fracture of coated materials.

References

1. ASTM E2734-04. Standard guide for acoustic emission system performance verification
2. ISO 13380-2002. Diagnostic of machine of using performance parameter. General guidelines
3. Abdulaziz, A., McCrory, J., Holford, K., Elsabbagh, A., Hedaya, M.: Experimental three-point bending test of glass fibre aluminium honeycomb sandwich panel with acoustic emission damage assessment. *Insight Nondestr. Test. Condition Monit.* **63**(12), 727–733 (2021). <https://doi.org/10.1784/insi.2021.63.12.727>
4. Al-Jumaili, S., Eaton, M., Holford, K., McCrory, J., Pullin, R.: Damage characterisation in composite materials under buckling test using acoustic emission waveform clustering technique. In: 53rd Annual Conference of The British Institute of Non-destructive Testing (2014)
5. Babichev, S., Škvor, J.: Technique of gene expression profiles extraction based on the complex use of clustering and classification methods. *Diagnostics* **10**(8) (2020). <https://doi.org/10.3390/diagnostics10080584>
6. Babichev, S., Lytvynenko, V., Skvor, J., Korobchynskiy, M., Voronenko, M.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction. In: Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, pp. 336–341 (2018). <https://doi.org/10.1109/DSMP.2018.8478452>
7. Babichev, S.A., Kornelyuk, A.I., Lytvynenko, V.I., Osypenko, V.V.: Computational analysis of microarray gene expression profiles of lung cancer. *Biopolymers Cell* **32**(1), 70–79 (2016). <https://doi.org/10.7124/bc.00090F>
8. Behnia, A., Chai, H., GhasemiGol, M., Sepehrinezhad, A., Mousa, A.: Advanced damage detection technique by integration of unsupervised clustering into acoustic emission. *Eng. Fract. Mech.* **210**, 212–227 (2019). <https://doi.org/10.1016/j.engfracmech.2018.07.005>
9. Buketov, A., Brailo, M., Yakushchenko, S., Saprionov, O., Smetankin, S.: The formulation of epoxy-polyester matrix with improved physical and mechanical properties for restoration of means of sea and river transport. *J. Mar. Eng. Technol.* **19**(3), 109–114 (2020). <https://doi.org/10.1080/20464177.2018.1530171>
10. Cho, H., Shoji, N., Ito, H.: Acoustic emission generation behavior in A7075-T651 and A6061-T6 aluminum alloys with and without cathodic hydrogen charging under cyclic loading. *J. Nondestr. Eval.* **37**(4), 1–7 (2018). <https://doi.org/10.1007/s10921-018-0536-7>
11. Dmitriev, A., Polyakov, V., Kolubaev, E.: Digital processing of acoustic emission signals in the study of welded compounds in metal alloys. *High-Perform. Comput. Syst. Technol.* **4**(1), 32–40 (2020)
12. Dmitriev, A., Polyakov, V., Lependin, A.: Investigation of plastic deformation of aluminum alloys using wavelet transforms of acoustic emission signals. *Russ. J. Nondestr. Test.* **8**(1), 33–36 (2018). <https://doi.org/10.22226/2410-3535-2018-1-33-36>
13. Dmitriev, A., Polyakov, V., Ruder, D.: Application of the principal component method to the study of acoustic emission signals in aluminum alloys. *News Altai State Univ. Phys.* **1**(99), 19–23 (2018). [https://doi.org/10.14258/izvasu\(2018\)1-02](https://doi.org/10.14258/izvasu(2018)1-02)
14. Fotouhi, M., Sadeghi, S., Jalalvand, M., Ahmadi, M.: Analysis of the damage mechanisms in mixed-mode delamination of laminated composites using acoustic emission data clustering. *J. Thermoplast. Compos. Mater.* **30**(3), 318–340 (2017). <https://doi.org/10.1177/0892705715598362>

15. Gagar, D., Foote, P., Irving, P.: Effects of loading and sample geometry on acoustic emission generation during fatigue crack growth: implications for structural health monitoring. *Int. J. Fatigue* **81**, 117–127 (2015). <https://doi.org/10.1016/j.ijfatigue.2015.07.024>
16. Hao, W., Huang, Z., Xu, Y., Zhao, G., Chen, H., Fang, D.: Acoustic emission characterization of tensile damage in 3D braiding composite shafts. *Polym. Test.* **81**, 106176 (2020). <https://doi.org/10.1016/j.polymertesting.2019.106176>
17. Harizi, W., Chaki, S., Bourse, G., Ourak, M.: Damage mechanisms assessment of glass fiber-reinforced polymer (GFRP) composites using multivariable analysis methods applied to acoustic emission data. *Compos. Struct.* **289**(1), 115470 (2022). <https://doi.org/10.1016/j.compstruct.2022.115470>
18. Hase, A.: Early detection and identification of fatigue damage in thrust ball bearings by an acoustic emission technique. *Lubricants* **8**(3), 37 (2020). <https://doi.org/10.3390/lubricants8030037>
19. He, M., He, D., Qu, Y.: A new signal processing and feature extraction approach for bearing fault diagnosis using ae sensors. *J. Fail. Anal. Prev.* **16**(5), 821–827 (2016). <https://doi.org/10.1007/s11668-016-0155-5>
20. Hongwu, Q., Chao, Z., Xian, Z., Qinyin, F.: Research of acoustic emission testing method with application to monitored for wind turbines. *Int. J. Multimedia Ubiquit. Eng.* **10**(1), 109–118 (2015)
21. Huang, J., Zhang, Z., Han, C., Yang, G.: Identification of deformation stage and crack initiation in tc11 alloys using acoustic emission. *Appl. Sci.* **10**(11), 3674 (2020). <https://doi.org/10.3390/app10113674>
22. Lavanya, S., Mahadevan, S., Mukhopadhyay, C.K., Kumar, S.A.: Acoustic emission during press-brake bending of ss 304L sheets and its correlation with residual stress distribution after bending. *J. Mater. Eng. Perform.* (5), 1–12 (2021). <https://doi.org/10.1007/s11665-021-06250-w>
23. Lependin, A.A., Polyakov, V.V.: Scaling of the acoustic emission characteristics during plastic deformation and fracture. *Tech. Phys.* **59**(7), 1041–1045 (2014). <https://doi.org/10.1134/S1063784214070184>
24. Li, D., Tan, M., Zhang, S., Ou, J.: Stress corrosion damage evolution analysis and mechanism identification for prestressed steel strands using acoustic emission technique. *Struct. Control Health Monit.* **25**(8), e2189 (2018). <https://doi.org/10.1002/stc.2189>
25. Louda, P., Marasanov, V., Sharko, A., Stepanchikov, D., Sharko, A.: The theory of similarity and analysis of dimensions for determining the state of operation of structures under difficult loading conditions. *materials. Materials* **15**(3), 1191 (2022). <https://doi.org/10.3390/ma15031191>
26. Louda, P., Sharko, A., Stepanchikov, D.: An acoustic emission method for assessing the degree of degradation of mechanical properties and residual life of metal structures under complex dynamic deformation stresses. *Materials* **14**(9), 2090 (2021). <https://doi.org/10.3390/ma14092090>
27. Lu, D., Yu, W.: Correlation analysis between acoustic emission signal parameters and fracture stress of wool fiber. *Text. Res. J.* **89**(21–22), 4568–4580 (2019). <https://doi.org/10.1177/0040517519838057>
28. Lytvynenko, V., Lurie, I., Krejci, J., Voronenko, M., Savina, N., Taif, M.A.: Two step density-based object-inductive clustering algorithm. In: *CEUR Workshop Proceedings*, vol. 2386, pp. 117–135 (2019)
29. Babichev, S., Lytvynenko, V. (eds.): *ISDMCI 2021. LNDECT*, vol. 77. Springer, Cham (2022). <https://doi.org/10.1007/978-3-030-82014-5>

30. Marasanov, V., Sharko, A.: Energy spectrum of acoustic emission signals in complex media. *J. Nano- Electro. Phys.* **9**(4), 04024-1–04024-5 (2017). [https://doi.org/10.21272/jnep.9\(4\).04024](https://doi.org/10.21272/jnep.9(4).04024)
31. Marasanov, V., Sharko, A.: The energy spectrum of the acoustic emission signals of nanoscale objects. *J. Nano-electron. Phys.* **9**(2), 02012-1–02012-4 (2017). [https://doi.org/10.21272/jnep.9\(2\).02012](https://doi.org/10.21272/jnep.9(2).02012)
32. Marasanov, V.V., Sharko, A.V., Sharko, A.A.: Boundary-value problems of determining the energy spectrum of acoustic emission signals in conjugate continuous media. *Cybern. Syst. Anal.* **55**(5), 851–859 (2019). <https://doi.org/10.1007/s10559-019-00195-8>
33. Babichev, S., Peleshko, D., Vynokurova, O. (eds.): DSMP 2020. CCIS, vol. 1158. Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-61656-4>
34. Marasanov, V., Stepanchikov, D., Sharko, O., Sharko, A.: Operator of the dynamic process of the appearance of acoustic emission signals during deforming the structure of materials. In: IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO), pp. 646–650 (2020). <https://doi.org/10.1109/ELNANO50318.2020.9088893>
35. Babichev, S., Lytvynenko, V., Wójcik, W., Vyshemyrskaya, S. (eds.): ISDMCI 2020. AISC, vol. 1246. Springer, Cham (2021). <https://doi.org/10.1007/978-3-030-54215-3>
36. Mengyu, C., Xinglong, H., Zaoxiao, Z., Quan, D.: Identification and prediction of fatigue crack growth under different stress ratios using acoustic emission data. *Int. J. Fatigue* **160**(106860) (2022). <https://doi.org/10.1016/j.ijfatigue.2022.106860>
37. Mengyu, C., Zaoxiao, Z., Quan, D.: A new qualitative acoustic emission parameter based on Shannon's entropy for damage monitoring. *Mech. Syst. Sig. Process.* **100**(1), 617–629 (2018). <https://doi.org/10.1016/j.ymsp.2017.08.007>
38. Mieza, J., Oliveto, M., López Pumarega, M., Armeite, M., Ruzzante, J., Piotrkowski, R.: Identification of ae bursts by classification of physical and statistical parameters. In: AIP Conference Proceedings, vol. 760, p. 1174 (2005). <https://doi.org/10.1063/1.1916805>
39. Muir, C., et al.: Damage mechanism identification in composites via machine learning and acoustic emission. *NPJ Comput. Mater.* **7**(95) (2021). <https://doi.org/10.1038/s41524-021-00565-x>
40. Nedoseka, A., Nedoseka, S., Markashova, L., Kushnareva, O.: On identification of structural changes in materials at fracture by acoustic emission data. *Tech. Diagn. Nondestr. Test.* (4), 9–13 (2016). <https://doi.org/10.15407/tdnk2016.04.02>
41. Roundi, W., El Mahi, A., El Gharad, A., Rebiere, J.L.: Acoustic emission monitoring of damage progression in glass/epoxy composites during static and fatigue tensile tests. *Appl. Acoust.* **132**, 124–134 (2018). <https://doi.org/10.1016/j.apacoust.2017.11.017>
42. Saidi, L., Ali, J., Bechhoefer, E., Benbouzid, M.: Wind turbine high-speed shaft bearings health prognosis through a spectral kurtosis-derived indices and SVR. *Appl. Acoust.* **120**, 1–8 (2017). <https://doi.org/10.1016/j.apacoust.2017.01.005>
43. Saponov, O., Buketov, A., Maruschak, P., et al.: Research of crack initiation and propagation under loading for providing impact resilience of protective coating. *Funct. Mater.* **26**(1), 114–120 (2019). <https://doi.org/10.15407/fm26.01.114>
44. Sharko, M., Gonchar, O., Tkach, M., et al.: Intellectual information technologies of the resources management in conditions of unstable external environment. In: Babichev, S., Lytvynenko, V. (eds.) ISDMCI 2021. LNDECT, vol. 1158, pp. 519–533. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-82014-5_35

45. Sharko, M., Gusarina, N., Petrushenko, N.: Information-entropy model of making management decisions in the economic development of the enterprises. In: Lytvynenko, V., Babichev, S., Wójcik, W., Vynokurova, O., Vysheymyrskaya, S., Radetskaya, S. (eds.) ISDMCI 2019. AISC, vol. 1020, pp. 304–314. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-26474-1_22
46. Tian, Y., Yu, R., Zhang, Y., Zhao, X.: Application of acoustic emission characteristics in damage identification and quantitative evaluation of limestone. *Adv. Eng. Sci.* **52**(3), 115–122 (2020). <https://doi.org/10.15961/j.jsuese.201900315>
47. Wuriti, G., Chattopadhyaya, S., Thomas, T.: Acoustic emission test method for investigation of m250 maraging steel pressure vessels for aerospace applications. In: *Materials Today: Proceedings*, vol. 49, pp. 2176–2182 (2022). <https://doi.org/10.1016/j.matpr.2021.09.087>
48. Zarif Karimi, N., Minak, G., Kianfar, P.: Analysis of damage mechanisms in drilling of composite materials by acoustic emission. *Compos. Struct.* **131**, 107–114 (2015). <https://doi.org/10.1016/j.compstruct.2015.04.025>
49. Özaslan, E., Yetgin, A., Acar, B., Güler, M.: Damage mode identification of open hole composite laminates based on acoustic emission and digital image correlation methods. *Compos. Struct.* **274**, 114299 (2021). <https://doi.org/10.1016/j.compstruct.2021.114299>
50. Zhang, Y., Luo, H., Li, J., Lv, J., Zhang, Z., Ma, Y.: An integrated processing method for fatigue damage identification in a steel structure based on acoustic emission signals. *J. Mater. Eng. Perform.* **26**(4), 1784–1791 (2017). <https://doi.org/10.1007/s11665-017-2616-8>
51. Zhang, Y., Zhou, B., Yu, F., Chen, C.: Cluster analysis of acoustic emission signals and infrared thermography for defect evolution analysis of glass/epoxy composites. *Infrared Phys. Technol.* **112**, 103581 (2021). <https://doi.org/10.1016/j.infrared.2020.103581>
52. Zhao, G., Zhang, L., Tang, C., Hao, W., Luo, Y.: Clustering of ae signals collected during torsional tests of 3D braiding composite shafts using PCA and FCM. *Compos. B Eng.* **161**, 547–554 (2019). <https://doi.org/10.1016/j.compositesb.2018.12.145>
53. Shen, G., Zhang, J., Wu, Z. (eds.): *WCAE 2017. SPP*, vol. 218. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-12111-2>
54. Zhou, W., Zhao, W., Zhang, Y., Ding, Z.: Cluster analysis of acoustic emission signals and deformation measurement for delaminated glass fiber epoxy composites. *Compos. Struct.* **195**, 349–358 (2018). <https://doi.org/10.1016/j.compstruct.2018.04.081>
55. Zou, S., Yan, F., Yang, G., Sun, W.: The identification of the deformation stage of a metal specimen based on acoustic emission data analysis. *Sensors* **17**(4), 789 (2017). <https://doi.org/10.3390/s17040789>



Computational Intelligence in Medicine

Oleh Berezsky , Oleh Pitsun ^(✉) , Petro Liashchynskiy , Bohdan Derysh ,
and Natalia Batryn 

West Ukrainian National University, Ternopil, Ukraine

{ob,o.pitsun,dbb,nbatryn}@wunu.edu.ua, p.liashchynskiy@st.wunu.edu.ua

Abstract. Two paradigms have historically formed in artificial intelligence: neurocybernetics and black box cybernetics. The cybernetics of the “black box” is based on a logical approach. The rapid development of modern medicine is due to the use of technical diagnostic tools, and the use of new information technologies. Technical means of diagnosing allow you to visualize the processes of diagnosing. Intelligent information technologies use the methods and tools of artificial intelligence and can speed up the diagnosis and improve its accuracy. The authors of the work have been working on biomedical images for many years, building CAD systems with elements of artificial intelligence. Biomedical (cytological, histological, and immunohistochemical) images are used for diagnosis in oncology. The problems of classification, generation, segmentation, and clustering of biomedical images are solved in the work. For these purposes, the following means of computational intelligence were used: CNN, GAN, and U-net. CNN is used in the paper to classify images. Several models for the classification of biomedical images are proposed and a comparative analysis with existing analogs is given. The accuracy of classification for cytological images was 86%, for histological was 84%. The authors analyzed the architectures of convolutional neural networks of the U-net type for automatic segmentation of immunohistochemical images. A modified neural network architecture for the segmentation of immunohistochemical images has been developed. Computer experiments were performed on different numbers of stages and iterations. ROC curves are built to assess the quality of segmentation of known and modified network architectures such as U-net.

Keywords: Convolutional neural networks · GAN · U-net · Breast cancer · Cytological images · Immunohistochemical images

1 Introduction

In 2020, breast cancer became the most common type of cancer in the world. According to estimates from the IARC, in 2020, there were more than 2.26 million new breast cancer cases and nearly 685,000 deaths worldwide [16]. Key factors in the fight against breast cancer are early diagnosis and effective treatment.

There are the following methods for the diagnosis of cancer: cytological, histological, and immunohistochemical [8]. The cytological method is based on the study of individual cells. It allows for detecting pathologies of individual cells and making a preliminary diagnosis based on this detection. Histological examination is more accurate than cytological and involves the analysis of a group of cells of a particular tissue. The immunohistochemical examination is aimed to specify the diagnosis and choose the individual treatment. This is done by applying biomarkers to cancer cells (Estrogen, Progesterone, Her2neu, KI-67).

Artificial intelligence computer systems are used to diagnose cancer. There are two paradigms that have historically been formed in artificial intelligence: neurocybernetics and black box cybernetics. Black box cybernetics is based on a logical approach. The logical approach uses the laws of human thinking and communication and is based on classical logic. Neurocybernetics uses the laws of the central and peripheral nervous systems. The results of this approach are artificial neural networks and artificial immune systems. Besides, artificial intelligence uses patterns of animal behavior, for example, ants, bees, etc. Therefore, ant colony algorithms and swarm algorithms appeared. The use of the achievements of the evolution of flora and fauna (natural selection) led to evolutionary computation and genetic algorithms. The development of classical logic was “fuzzy” logic, which is a characteristic of human thinking. Thus, the paradigm of neurocybernetics gave rise to a separate area of artificial intelligence - computational intelligence. This approach was formed as a result of integration and further development of neural networks, learning, self-learning, evolutionary modeling of genetic algorithms, fuzzy logic systems, and fuzzy neural networks.

Computational intelligence is used in pattern recognition, image classification, medical and technical diagnostics, and natural and economic forecasting. The use of technical diagnostic tools and new information technologies contributed to the rapid development of modern medicine. Technical diagnostic tools make it possible to visualize diagnostic processes. Intelligent information technologies use methods and tools of artificial intelligence and make it possible to speed up the diagnosis and increase its accuracy. In general, medicine is an inaccurate science that is highly dependent on subjective factors. Therefore, “fuzzy” logic is widely used in medicine. The authors of the research study have been working on biomedical image processing and the development of artificial intelligence CAD systems for many years. Biomedical (cytological, histological, and immunohistochemical) images have been used for diagnosis in oncology.

The authors aimed to solve the problems of biomedical image classification, generation, and segmentation. The following means of computational intelligence were used for these purposes: CNN, GAN, and U-net.

2 Problem Statement

Extensive analysis of recent scientific articles showed that the methods of computational intelligence are widely used in medicine. Therefore, this area of research is relevant. The problem lies in effective use of computational intelligence methods for the analysis of cytological, histological, and immunohistochemical images.

In the study, the authors present the results of research on immunohistochemical image segmentation, classification and generation based on the use of deep learning tools: U-net, CNN, and GAN.

3 Literature Review

The purpose of the literature review is to analyze the recent research studies related to computational intelligence in medicine.

The researchers investigated various areas of applications of computational intelligence in medicine. For example, Peng, Yao, and Jiang (2006) presented a new approach to breast cancer detection based on a genetic algorithm (GA) [39]. In the proposed approach, the genetic algorithm is used to search for bright spots on the mammogram. De Oliveira Silva, Barros, and Lopes [38] used the independent component analysis to detect lesions in dense breasts and used an area overlay measure to verify the results. Some researchers proposed algorithms for classifying cell stains (benign and cancerous). It was achieved by combining outputs of ResNets' processed images in different magnification factors using a meta-decision tree [20]. Others investigated batch mode active learning on the Riemannian manifold for automated scoring of nuclear pleomorphism [18]. The authors stated that the best method is active learning. Besides, there are many review articles, in which the authors analyzed the recent literature in the area. For example, Kazeminia et al. considered the most recent and relevant publications on GANs for medical applications [28].

Also, Ghasemi et al. [21] presented a new approach to the classification of medical images. In their research study, de Almeida Thomaz, Sierra-Franco, and Raposo [3] developed an algorithm for selecting the best area in the image to obtain a polyp. The proposed algorithm increases the variability of data using a generative network. Furthermore, researchers proposed BCHisto-Net, which is based on local and global features of the breast histopathological image classification at 100x magnification [42].

In their article, Abd Latif and Faye [30] presented a method of automatic segmentation - automated tibiofemoral joint segmentation based on a deeply supervised 2D-3D ensemble U-Net. Other researchers proposed weak label-based Bayesian U-Net exploiting Hough transform-based annotations to segment optic discs in fundus images [45]. Besides, some researchers also investigated methods for detecting and treating breast cancer with the help of deep learning [35]. Others explored the ways in which deep learning can be best integrated into breast cancer detection using artificial intelligence techniques [6].

The researchers also explored classical algorithms of image classification. For example, the algorithms' hyperparameters optimization was carried out. The results have been compared with related works [24]. Kumar et al. [29] proposed a systematic review of artificial intelligence techniques in cancer prediction and diagnosis. Also, the authors analyzed the role of artificial neural networks (ANN) in cytopathology. The vast majority of CNS applications are related to cervical cytopathology [41].

In their research studies, some researchers provided an extensive analysis of the methods used to detect and diagnose breast cancer [2]. Others investigated fibroadenoma. The experiments were performed using the k-means-based hybrid segmentation method [19]. Commercial systems for automation in cytology were investigated by Mitra et al. [34]. Deep learning algorithms for breast cancer diagnosis were developed by Jain et al. [25]

The use of neural networks for computer medical diagnostics (CAMD) in the last decade was investigated by Vasilakos et al. [26] De Matos et al [32] presented an overview of machine learning methods for histopathological image analysis, including methods of shallow and deep learning.

Olatunji et al. [37] developed classification tools based on machine learning. The techniques used in this research study include random forest (RF), artificial neural network (ANN), support vector machine (SVM), and Naive Bayes (NB). **The objective of the research** is to compare various internal clustering quality criteria calculated based on the use of IMOC technique with following calculation of both the external and balance criteria for clustering various types of two dimensional synthetic datasets applying density based DBSCAN clustering algorithm.

4 Immunohistochemical Image Analysis

Immunohistochemical research studies play an important role in differential diagnosis and accurate diagnosis [36]. Immunohistochemistry is an analytical method for determining proteins in biological tissue cells based on the antigen-antibody reaction [43]. Immunohistochemical studies are used to determine the expression of estrogen and progesterone receptors in breast cancer in order to define the further treatment tactics [46].

During the examination, biomarkers were used for the diagnosis. There are several biomarkers that are used for immunohistochemical images. The reaction of the immunohistochemical image to estrogen is shown in Fig. 1. The reaction of the immunohistochemical image to progesterone is shown in Fig. 2. The reaction of the immunohistochemical image to the biomarker HER-2/neu is shown in Fig. 3. The reaction of the immunohistochemical image to the biomarker Ki-67 is shown in Fig. 4.

The reaction of the immunohistochemical image to each biomarker is stored as a digital image. To diagnose, it is necessary to calculate the area of stained cell nuclei and their brightness. Based on these two indicators, a decision is made to assign the image to a certain class of diseases. Figure 5 shows the regions on the immunohistochemical image that are taken into account in the diagnosis.

In the subsequent stages of processing with the help of computer vision algorithms, the calculation of the area of cells and their average brightness level is carried out.

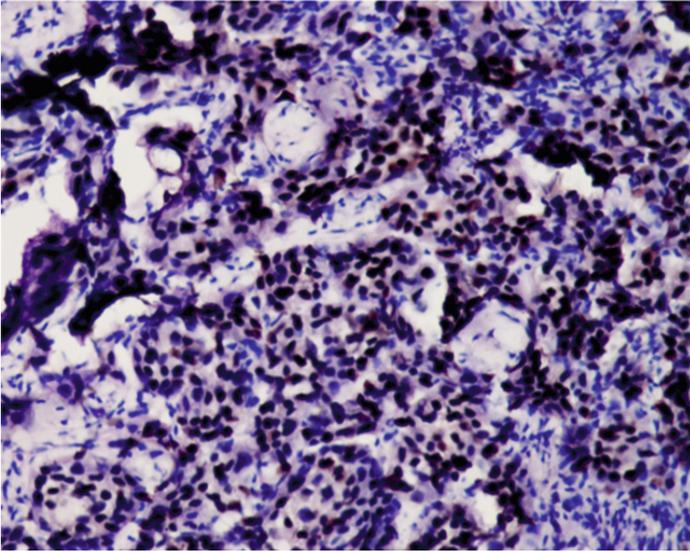


Fig. 1. Reaction to the Estrogen receptors

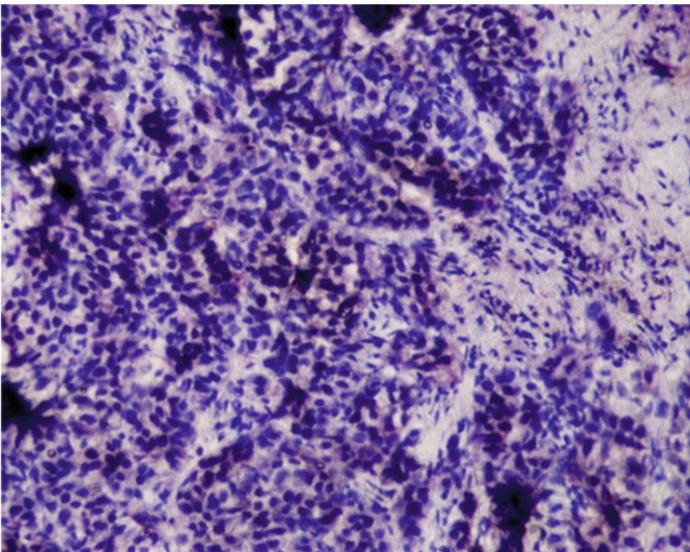


Fig. 2. Reaction to the Progesterone receptors

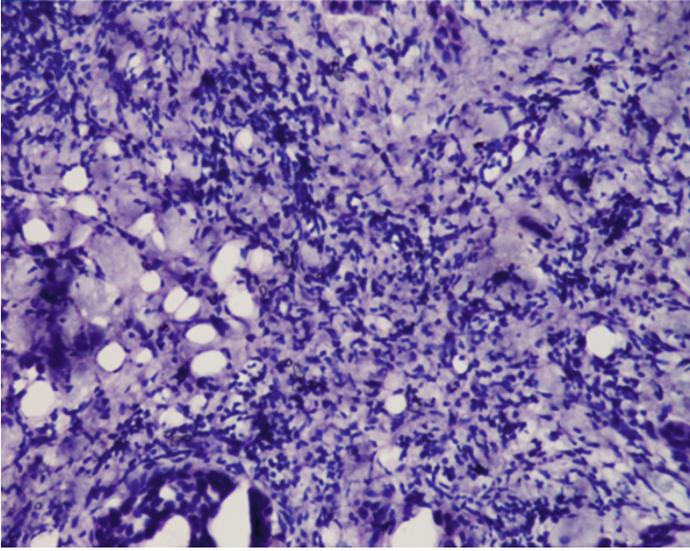


Fig. 3. Reaction to the biomarker HER-2/neu

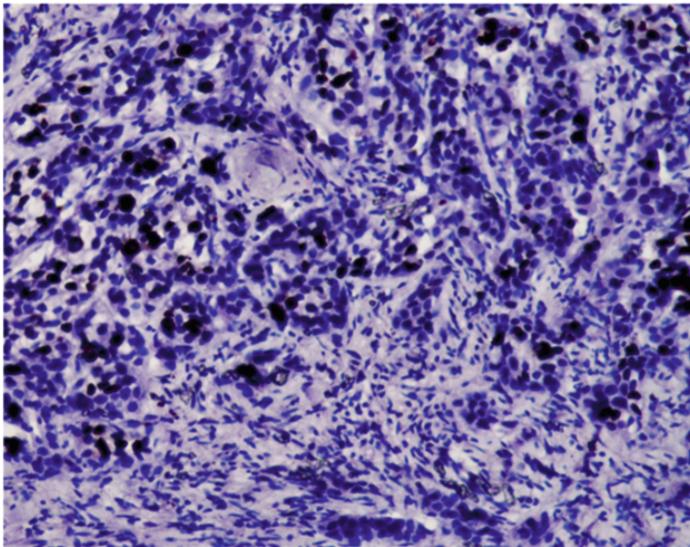


Fig. 4. Reaction to the biomarker Ki-67

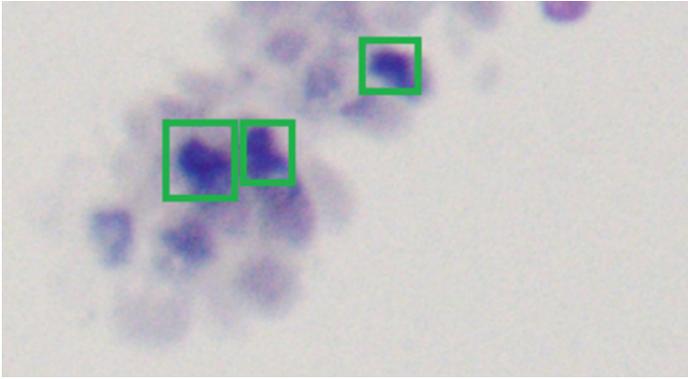


Fig. 5. Regions of interest considered for immunohistochemical image analysis

4.1 Image Segmentation

Image segmentation is one of the main stages of image analysis. Diagnosis based on immunohistochemical images involves the analysis of the cell nuclei size and the degree of their staining. Segmentation distinguishes image micro-objects from an uninformative background. In medicine, segmentation is used to detect tumors and other pathologies, diagnose, determine tissue volume, study the anatomical structure, and provide treatment planning. There are the following modern methods of segmentation:

- clustering methods;
- histogram methods;
- edge selection methods;
- building area methods;
- graph cut methods;
- watershed segmentation.

The U-net network is a type of convolutional neural network that has been developed for image segmentation. The network consists of two parts, that is, contraction path and expansion path, which give it a U-shaped architecture.

The examples of U-net architecture based neural network applications are the following:

- BRATS brain image segmentation [33].
- liver image segmentation “siliver07” [31].

There is no universal convolutional neural network architecture for segmentation. Therefore, researchers have developed many variations of U-net based neural network architectures. A variety of approaches to U-net segmentation was explored in [4]. The analysis of traditional approaches and methods of deep learning for immunohistochemical image segmentation was carried out in [40]. Benny

et al. in [7] conducted a comparative analysis of approaches to the immunohistochemical image segmentation for the breast cancer diagnosis. The authors concluded that the U-net network shows the best results. A software application for tissue segmentation based on the U-net architecture was discussed in [1]. A modification of the traditional U-net network for tumor detection was developed by [47]. The paper [13] presented a comparative analysis of U-net-based convolutional neural network architectures.

4.2 Software Structure Module for Automatic Image Segmentation

The U-net-based software structure module for automatic image segmentation is shown in Fig. 6.

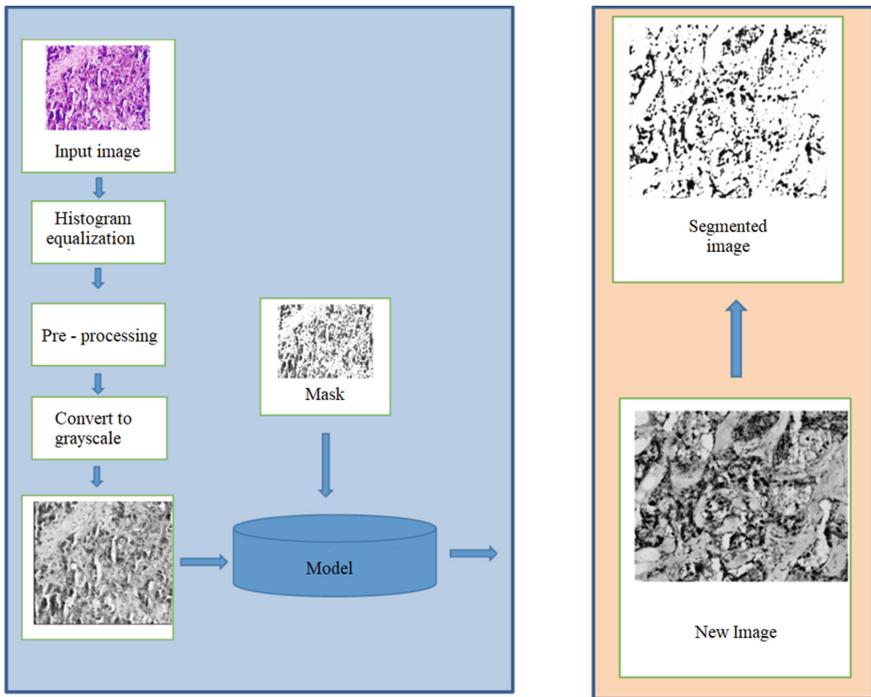


Fig. 6. Software structure module for automatic immunohistochemical image segmentation

For CNN-based systems, it is necessary to provide two stages: learning and network operation. The learning phase includes loading the input image into memory, performing histogram alignment, and pre-processing to reduce noise levels and histogram alignment. After the histogram alignment, the image is reformatted to grayscale.

Input images and their masks are input to the neural network for learning. Masks are made by hand. After learning, a neural network model is formed. At the operation stage, a new image from the test sample (without a mask) is received. The image interacts with a pre-trained neural network model. After the input image processing, a segmented image is generated and stored on the disk.

The listing of the automatic image correction code is the following:

```
public Mat autoImageCorrection(Mat src, int lowLevel){
    FiltersOperations filtoperation = new FiltersOperations
(src, "4", "5", "", "", "");
    Mat brightMat = PreProcImgOperations.
bright(filtoperation.getOutputImage(), 8);
    filtoperation.getOutputImage().release();
    Mat contrastMat = PreProcImgOperations.
contrast(brightMat, 2.0);
    brightMat.release();
    Mat result = Segmentation.watershed(contrastMat, lowLevel);
    Highgui.imwrite( "E:\\test\\result.jpg", result);

    return result;
}
```

The developed project was realized by using the Java programming language and the OpenCV computer vision library.

4.3 U-net Architecture for Immunohistochemical Image Segmentation

Figure 7 shows the developed structure of the neural network U-net encoder.

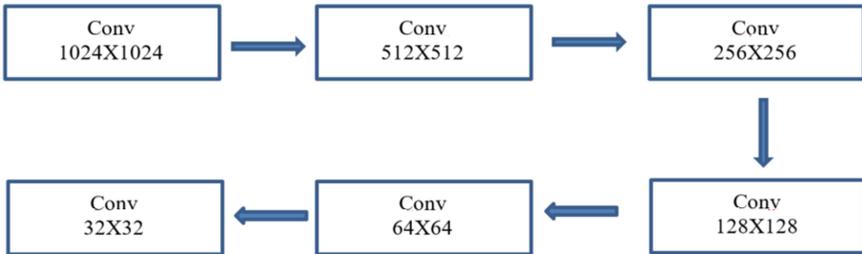


Fig. 7. Structure of the neural network U-net encoder

The neural network consists of a contraction path and expansion path, which give it a U-shaped architecture. Contraction is a typical convolutional network, which consists of the re-application of convolutions, each of which is followed

by the ReLU activation function and the max-pooling operation. To simplify the understanding of the encoder structure, Fig. 7 shows only the convolution operations, because it is the sequence of these operations that distinguishes the architectures of different U-net networks.

5 Computer Experiments

In the research study, an immunochemical image dataset was used. It consists of the following types:

- the result of the study (estrogen);
- the result of the study (progesterone);
- the result of the study (oncoprotein);
- the result of the study (cell proliferation).

Automatic segmentation quality has been assessed by computing the individual pixels between the received and original images. The listing of segmentation quality computation code is the following:

```

for (int i =0; i < pixelValues.size(); i++) {
    System.out.println("original" + pixelValues.get(i).Value
+ " generated " + generatedPixelValues.get(i).Value);
    if ((pixelValues.get(i).Value > 0 &&
generatedPixelValues.get(i).Value > 0)
|| (pixelValues.get(i).Value == 0 &&
generatedPixelValues.get(i).Value == 0)) {
        TP++;
    }else {
        TN++;
    }
    if (pixelValues.get(i).Value > 0
&& generatedPixelValues.get(i).Value > 0){
        bb++;
    }
    if (pixelValues.get(i).Value == 0
&& generatedPixelValues.get(i).Value == 0){
        ww++;
    }
    if (generatedPixelValues.get(i).Value > 0
&& pixelValues.get(i).Value == 0){
        bw++;
    }
    if (generatedPixelValues.get(i).Value == 0
&& pixelValues.get(i).Value > 0){
        wb++;
    }
}

```

Figure 8 shows the segmentation accuracy depending on the number of iterations for the architecture illustrated in Fig. 7. Accuracy is determined by calculating the number of correctly classified pixels to the total number of pixels.

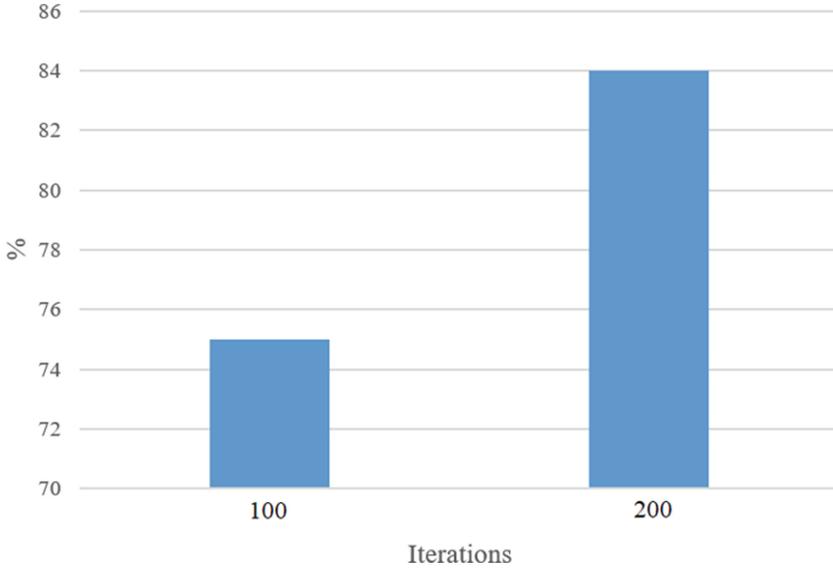


Fig. 8. Segmentation accuracy depending on the number of iterations

6 Image Generation

There are several well-known approaches to image synthesis. One of the approaches is based on the synthesis of whole fragments, pixel-by-pixel and combined. Another approach is procedural, which is based on a formal description of the synthesis procedure. After the development of GAN [22], the neural network became a popular image synthesis approach. The GAN approach to biomedical image synthesis has become especially widespread. This is due to the objectively small biomedical image samples.

Let us consider the basic GAN architectures.

DCGAN. The architecture of this network is based on the usual GAN architecture; however, it uses the convolution layer instead of the fully connected layer in the generator and discriminator. This allows for increasing the stability of learning [5]. The following loss function $V(G, D)$ is used to assess the quality of training:

$$\min_G \max_D V(G, D) = E_x P_{data}(X) [\log D(x)] + E_z p_z(z) [\log(1 - D(G(z)))] \quad (1)$$

where G can be a generator, a discriminator, and a mathematical expectation.

WGAN. In this architecture, a discriminator is replaced with a critic. The critic does not compute the probability of an image as being real or synthesized.

The critic shows how different a synthesized image is from a real one. So, this loss function is a metric that shows the difference between a synthesized image and a real image [17].

This architecture also applies a constraint of discriminator gradient to prevent so-called gradient attenuation. The values that constrain the gradients are called hyperparameters. Hyperparameters are set before the training.

Apart from the loss function and the gradient constraint method, this architecture does not differ from the DCGAN architecture.

BigGAN. This architecture was designed to synthesize higher-resolution images. This is achieved by using models with more layers and parameters. The main differences are in the use of larger batches and “Large Scale GAN”.

The architecture is based on the SAGAN (Self-Attention GAN) model. The main concept of the architecture is the self-attention mechanism, which allows the generator and the discriminator to focus on different parts of the image. These parts of the image are dependent on one context [48]. The technique allows GAN to synthesize complex textures where there are many dependent details.

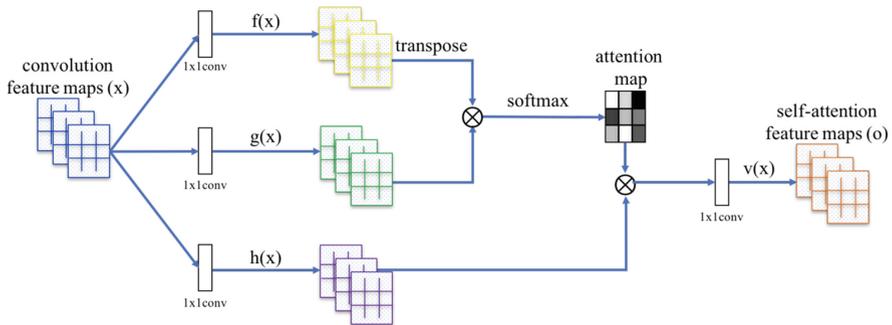


Fig. 9. Self-attention mechanism

Another innovation is spectral normalization. It is possible to draw an analogy with the gradient constraint in the WGAN architecture, but the constraint maps of convolutional layers are limited. This approach significantly reduces the speed of learning. Therefore, researchers often use a double update rule, where the discriminator parameters are updated twice before each generator parameter update.

ContraGAN. This architecture is based on BigGAN but uses a different strategy to generate label-based images. All previous architectures that allow label-based imaging use the Conditional GAN, which involves adding a class label

to the generator and discriminator models along with the image. These architectures use data-to-class relationships between images and labels. The authors of ContraGAN suggest using data-to-class and data-to-data relationships [27]. Ultimately, the user will be able to generate images of the classes that trained the network simply bypassing the class label and noise vector to the input of the generator.

The peculiarity of this architecture is that the discriminator maximizes the mutual information between the samples to represent the same class. Thus, the generator tries to maximize information between the synthesized images of one class.

Dataset. A sample of cytological images of 64 by 64 pixels was selected for the experiments. This sample was expanded by applying a set of affine transformations (random rotation, shear, scaling, and reflection).

Hardware. The IBM virtual machine with the following configuration was used for the experiments: 60 GB RAM, 8 vCPU x 2GHz, Nvidia Tesla P100 GPU 16 GB. The experiments were performed using the following GAN network architectures: DCGAN, WGAN, and BigGAN. Each of the networks was trained for 30,000 iterations [9].

FID and Inception Score metrics were used to assess the quality of the synthesized images. Inception Score (IS) is a metric for assessing image quality based on the Google Inception 3 model. Images are classified by model, which allows determining the probability that each image belongs to a certain class. Based on these probabilities, the IS value is calculated [44]. The metric has the smallest value, which is equal to 1, and the largest, which is equal to the number of classes. In this case, Google Inception 3 is trained on the ILSVRC 2012 dataset, which has 1000 classes. Therefore, the largest IS value will be 1000 for this dataset. The higher the value of the metric, the better it is. FID (Frechet Inception Distance) is a metric that calculates the distance between the feature vectors of real images and synthesized images [23]. Besides, the metric is also based on the Google Inception 3 model and is an improved version of the Inception Score (IS) metric.

The results of the experiments are shown in the Table 1.

Table 1. The results of the experiments

Network	Training time	FID	IS.	Precision	Recall	Density	Coverage.
DCGAN	13 hr. 40 min.	147,88	3,30	0,05	0,0025	0,013	0,01
WGAN	7 hr.	360,74	1,32	0	0	0	0
BigGAN	11 hr. 45 min.	300,14	1,85	0	0	0	0

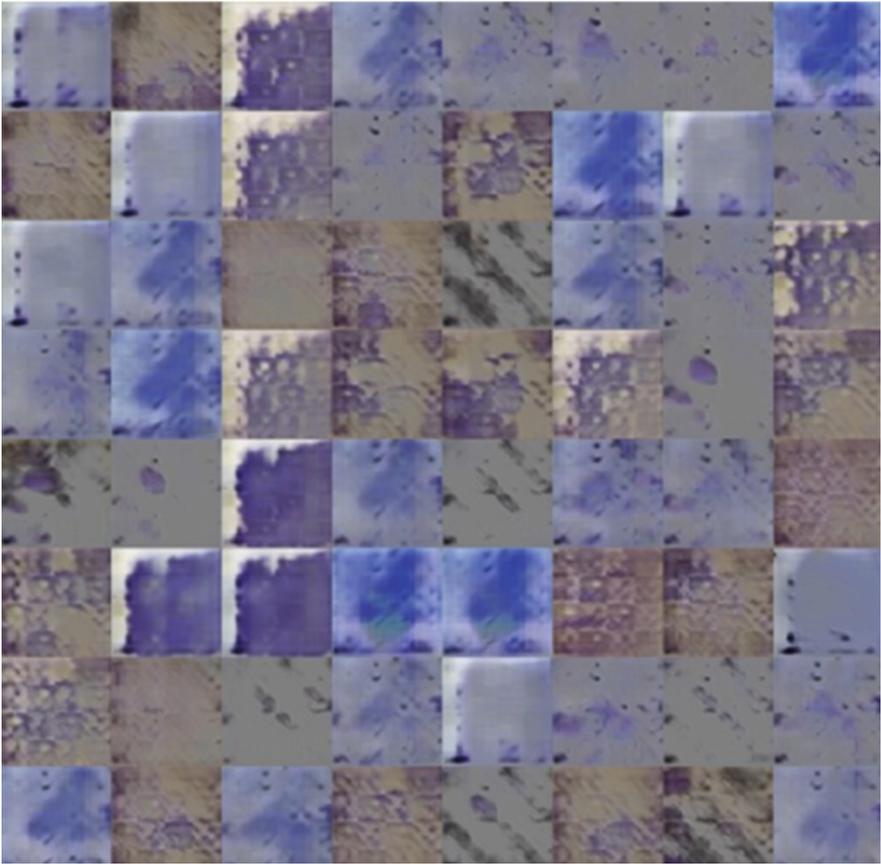


Fig. 10. DCGAN synthesized images

Besides, a separate experiment was conducted using the ContraGAN network, which was trained for 100,000 iterations.

The generated images were added to the original sample. The training of the classifier on the original sample gave a classification accuracy of about 76%. The training of the classifier on the extended sample gave a classification accuracy of about 97%. ROC curves are shown in the figures.

7 Cytological Image Classification

The use of artificial neural networks and deep learning methods became a current trend in image analysis. Convolutional neural networks (CNN) are increasingly often used to classify images, particularly in medicine. CNN networks use a variety of multilayer perceptrons. CNN can reduce the amount of information

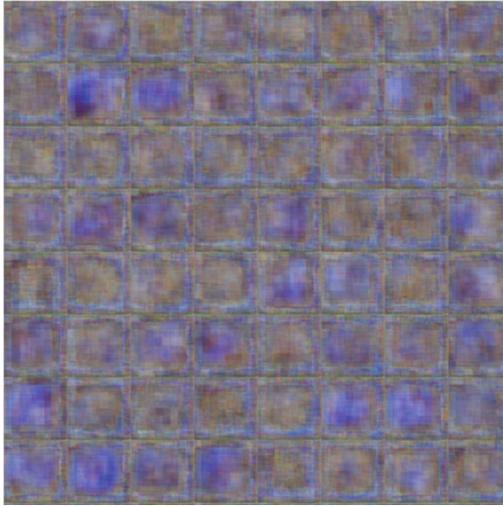


Fig. 11. WGAN synthesized images

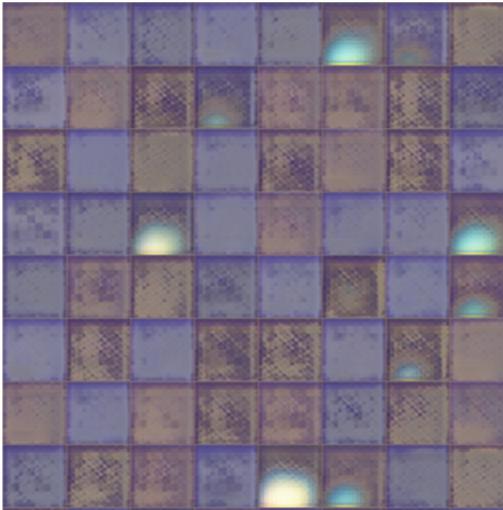


Fig. 12. BigGAN synthesized images

Table 2. ContraGAN configuration

Network	Training time	FID	IS.	Precision	Recall
ContraGAN	2 days 4 h	24,77	3,57	0,97	0,97

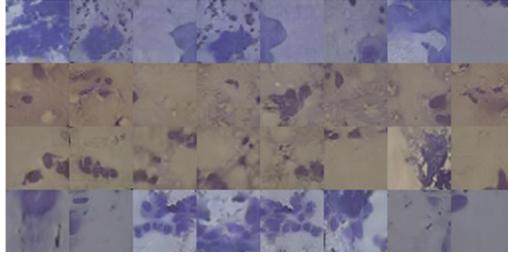


Fig. 13. ContraGAN synthesized images

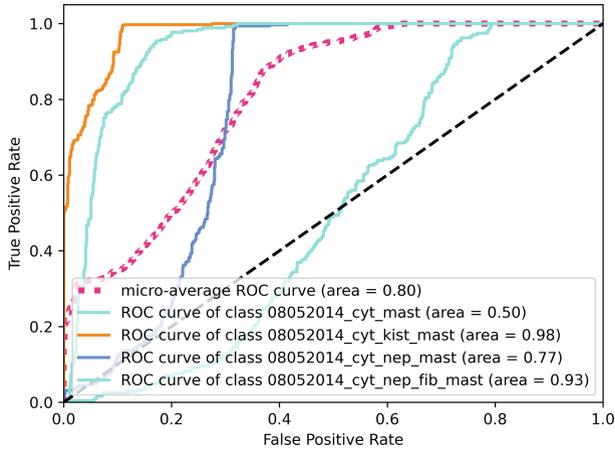


Fig. 14. ROC curve for the classifier (original sample)

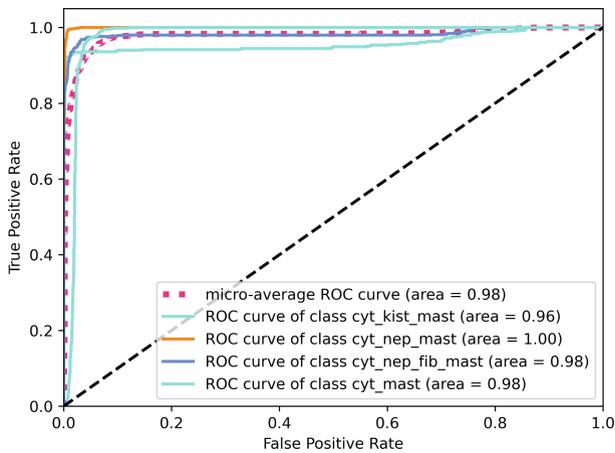


Fig. 15. ROC curve for the classifier (extended sample)

by means of using special operations (convolutions). These networks work better with higher-resolution images and allow highlighting image features such as edges, contours, or facets.

The authors of the research study a variety of biomedical images, such as cytological images. The database of cytological images used for classification was divided into 4 classes. An example of cytological images is shown in Fig. 16.

To improve image classification accuracy, it is necessary to carry out pre-processing [10]. Filtration was used as a pre-treatment. Filtration is usually used to reduce noise levels, including pulsed noise. Brightness adjustments are used to get clearer contours of the image micro-objects. The following parameters of the median filter window have been established by experimental studies:

$$\begin{cases} mw = 5 \times 5, PSNR < 20dB \\ mw = 3 \times 3, PSNR > 20dB. \end{cases} \quad (2)$$

where: mw is a size of the median filter window; PSNR is Peak Signal-to-Noise Ratio.

After improving the image quality, it is necessary to reduce and cluster the data [14, 15].

An important characteristic of a convolutional neural network is architecture. Today, there are a sufficient number of convolutional neural network architectures that have been developed for special types of images: X-ray images, MRI exams, etc.

Figures 17, 18 and 19 shows the architectures of convolutional neural networks used in this study.

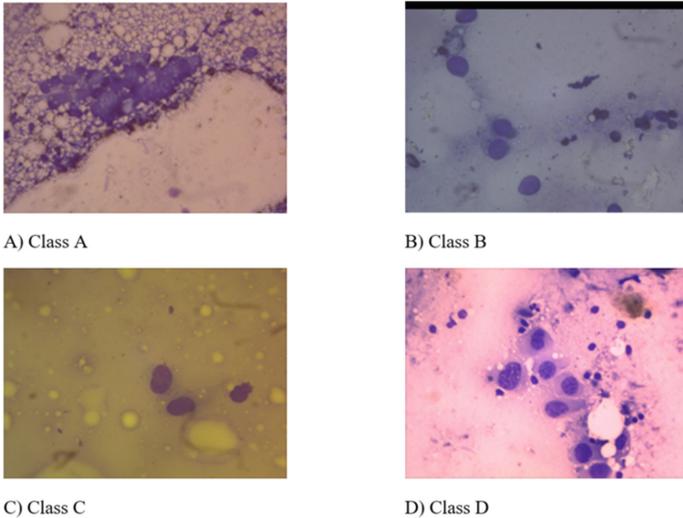


Fig. 16. Examples of cytological images



Fig. 17. CNN architecture (Type 1)



Fig. 18. CNN architecture (Type 2)



Fig. 19. CNN architecture (Type 3)

When working with convolutional neural networks, the number of epochs is an important parameter that affects the quality and duration of learning. Figure 20 shows the results of classification accuracy with the number of epochs = 10.

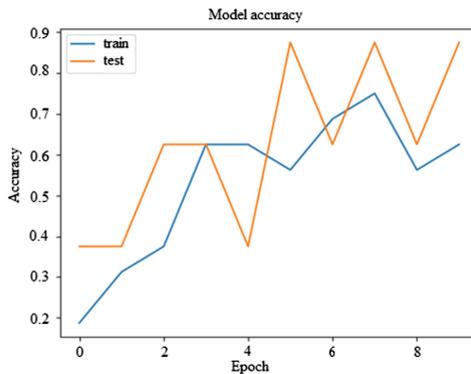


Fig. 20. Comparative analysis of the classification accuracy depending on the number of epochs

Figure 21 shows the results of classification accuracy with the number of epochs = 20.

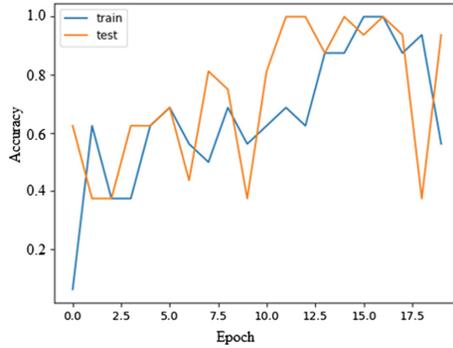


Fig. 21. Comparative analysis of the classification accuracy depending on the number of epochs

In this example, the accuracy of the test sample classification is 87% for 10 epochs and 91% for 20 epochs.

Figure 22 shows the accuracy of the classification of the training and test samples for architecture (Fig. 19).

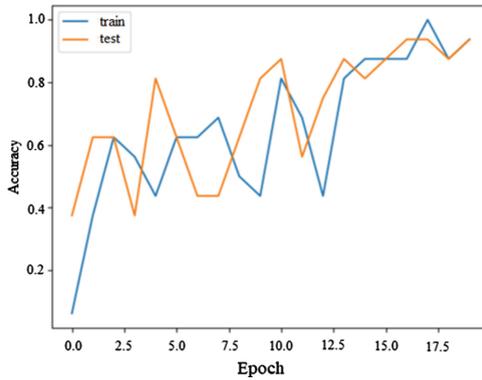


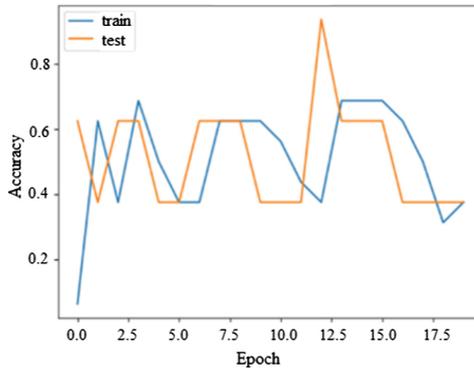
Fig. 22. Accuracy of the classification of the training and test samples for architecture (Fig. 19)

Table 3 shows a comparative analysis of the accuracy of the test sample classification for convolutional neural network architectures.

Table 3. Comparative analysis of the accuracy of the test sample classification for CNN architectures

CNN architecture	Figure 17	Figure 18	Figure 19
Classification accuracy, %	87	65	91

The advantage of using convolutional neural networks is the use of RGB images. It significantly increases the accuracy of classification. For example, Fig. 23 shows the classification accuracy using a single channel (grayscale) image.

**Fig. 23.** Classification accuracy using a single channel (grayscale) image (Color figure online)

Analysis of Fig. 23 shows that the classification accuracy of the training and test samples is less than 50%. Computer experiments were conducted in the Python programming language using the Tensorflow and Keras libraries. The operating system was Linux Mint.

On the basis of the proposed methods of image classification, the authors developed algorithms for automatic diagnosis and modern automated microscopy techniques in oncology [11, 12].

8 Conclusions

Based on the analysis of the literature review, the authors showed the relevance of the use of methods and tools of artificial intelligence in medicine. In the research the authors analyzed biomedical images (immunohistochemical images) used for breast cancer analysis. They have developed a U-net architecture, which showed the accuracy of histological image segmentation of 82% with a number of iterations equal to 200. As a result of synthesis, about 800 cytological images from each class were generated. After training the classifier on the original sample, the

classification accuracy was about 76%. Training the classifier on the extended sample gave a classification accuracy of 97%.

The CNN architecture for the classification of cytological images has been developed. It was found that this neural network architecture showed the results of 91% for the test sample.

Future research directions include the development and optimization of new CNN GAN U-net architectures for the synthesis, classification, and segmentation of cytological, histological, and immunohistochemical images used for diagnosis in oncology. Based on the developed methods of biomedical image classification, generation, and segmentation, it is planned to develop CAD systems with elements of artificial intelligence.

References

1. Abraham, R., Durkee, M.: Application and generalizability of U-Net segmentation of immune cells in inflamed tissue. In: Tomaszewski, J.E., Ward, A.D. (eds.) *Medical Imaging 2021: Digital Pathology*. SPIE, Online Only, United Stat, p. 40 (2021)
2. Al-Tam, R., Narangale, S.: Breast cancer detection and diagnosis using machine learning: a survey. *J. Sci. Res.* **65**(5), 265–285 (2021)
3. de Almeida Thomaz, V., Sierra-Franco, C., Raposo, A.: Training data enhancements for improving colonic polyp detection using deep convolutional neural networks. *Artif. Intell. Med.* **111**, 101988 (2021). <https://doi.org/10.1016/j.artmed.2020.101988>
4. Alom, M., Yakopcic, C., Taha, T.: Nuclei segmentation with recurrent residual convolutional neural networks based U-Net (R2U-Net), pp. 228–233 (2018)
5. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN (2017)
6. Bai, J., Posner, R., Wang, T.: Applying deep learning in digital breast tomosynthesis for automatic breast cancer detection: a review. *Med. Image Anal.* **71**, 102049 (2022). <https://doi.org/10.1016/j.media.2021.102049>
7. Benny, S., Varma, S.: Semantic segmentation in immunohistochemistry breast cancer image using deep learning. In: *2021 International Conference on Advances in Computing, Communication, and Control (ICAC3)*, pp. 1–2. IEEE, Mumbai, India (2021)
8. Berezsky, O., Batko, Y., Berezka, K., Verbovyi, S.: Methods, algorithms and software for processing biomedical images. *Ekonomichna dumka*, p. 330 (2017)
9. Berezsky, O., Liashchynskiy, P.: Comparison of generative adversarial networks architectures for biomedical images synthesis. *AAIT* **4**, 250–260 (2021). <https://doi.org/10.15276/aait.03.2021.4>
10. Berezsky, O., Pitsun, O., Derish, B.: Adaptive immunohistochemical image pre-processing method. In: *International Conference on Advanced Computer Information Technologies (ACIT)*, pp. 820–823. IEEE, Deggendorf, Germany (2020)
11. Berezsky, O., Pitsun, O., Batryn, N.: Modern automated microscopy systems in oncology. In: *Proceedings of the 1st International Workshop on Informatics and Data-Driven Medicine*, Lviv, Ukraine, pp. 311–325 (2018)
12. Berezsky, O., Pitsun, O., Datsko, T.: Specified diagnosis of breast cancer on the basis of immunogistochemical images analysis. In: *Proceedings of the 3rd International Conference on Informatics and Data-Driven Medicine*, CEUR, Växjö, Sweden, pp. 129–135 (2020)

13. Berezhsky, O., Pitsun, O., Derysh, B.: Automatic segmentation of immunohistochemical images based on U-Net architectures. In: CEUR, pp. 22–33 (2021). <http://ceur-ws.org/Vol-3038>
14. Berezhsky, O., Pitsun, O., Dolyniuk, T.: Cytological image classification using data reduction. In: Proceedings of the 2nd International Workshop on Informatics and Data-Driven Medicine (IDDM 2019), CEUR, Lviv, Ukraine, pp. 16–29 (2019)
15. Berezhsky, O., Pitsun, O., Dolyniuk, T.: Cytological images clustering of breast pathologies. In: PIEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), pp. 62–65. IEEE, Zbarazh, Ukraine (2020)
16. IARC, I.: Marks Breast Cancer Awareness Month 2021. <http://www.iarc.who.int/news-events/iarc-marks-breast-cancer-awareness-month-2021> (2022)
17. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis (2019)
18. Das, A., Nair, M.: Batch mode active learning on the Riemannian manifold for automated scoring of nuclear pleomorphism in breast cancer. *Artif. Intell. Med.* **103**, 101805 (2018). <https://doi.org/10.1016/j.artmed.2020.101805>
19. Filipczuk, P., Kowal, M., Obuchowicz, A.: Breast fibroadenoma automatic detection using k-means based hybrid segmentation method. In: 9th IEEE International Symposium on Biomedical Imaging (ISBI), pp. 1623–1626. IEEE, Barcelona, Spain (2012)
20. Gandomkar, Z., Brennan, P., Mello-Thoms, C.: MuDeRN: multi-category classification of breast histopathological image using deep residual networks. *Artif. Intell. Med.* **88**, 14–24 (2018). <https://doi.org/10.1016/j.artmed.2018.04.005>
21. Ghasemi, M., Kelarestaghi, M., Eshghi, F.: FDSR: a new fuzzy discriminative sparse representation method for medical image classification. *Artif. Intell. Med.* **106**, 101876 (2020). <https://doi.org/10.1016/j.artmed.2020.101876>
22. Goodfellow, I., Pouget-Abadie, J., Mirza, M.: Generative adversarial networks (2014)
23. Heusel, M.: GANs trained by a two time-scale update rule. <http://arxiv.org/abs/1706.08500> (2017)
24. Hlavcheva, D., Yaloveha, V., Podorozhniak, A.: A comparison of classifiers applied to the problem of biopsy images analysis. *AIS* **4**, 12–16 (2022). <https://doi.org/10.20998/2522-9052.2020.2.03>
25. Jain, P., Kulsum, S.: Analysis of deep learning algorithms for breast cancer diagnosis on WBC. *Int. J. Eng. Res. Technol.* **8**(15) (2020)
26. Jin, D., Vasilakos, A.: Neural networks for computer-aided diagnosis in medicine: a review. *Neurocomputing* **216**, 700–708 (2016). <https://doi.org/10.1016/j.neucom.2016.08.039>
27. Kang, M., Park, J.: ContraGAN: Contrastive learning for conditional image generation (2021)
28. Kazemina, S., Baur, C., Kuijper, A.: GANs for medical image analysis. *Artif. Intell. Med.* **109**, 101938 (2020)
29. Kumar, Y., Gupta, S., Singla, R., Hu, Y.-C.: A systematic review of artificial intelligence techniques in cancer prediction and diagnosis. *Arch. Comput. Methods Eng.* **29**(4), 2043–2070 (2021). <https://doi.org/10.1007/s11831-021-09648-w>
30. Latif, M., Faye, I.: Automated tibiofemoral joint segmentation based on deeply supervised 2D-3D ensemble U-Net: Data from the osteoarthritis initiative. *Artif. Intell. Med.* **122**, 102213 (2021). <https://doi.org/10.1016/j.artmed.2021.102213>
31. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. [arXiv:1411.4038](https://arxiv.org/abs/1411.4038) (2015)

32. de Matos, J., Ataky, S., de Souza Britto, A.: Machine learning methods for histopathological image analysis: a review. *Electronics* **562**(10), 700–708 (2021). <https://doi.org/10.3390/electronics10050562>
33. Menze, B., Jakab, A., Bauer, S.: The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging* **34**, 1993–2024 (2015). <https://doi.org/10.1109/TMI.2014.2377694>
34. Mitra, S., Das, N., Dey, S.: Cytology image analysis techniques towards automation: systematically revisited. *arXiv preprint arXiv:2003.07529* (2020)
35. Nassif, A., Talib, M.: Breast cancer detection using artificial intelligence techniques: a systematic literature review. *Artif. Intell. Med.* **127**, 102276 (2022). <https://doi.org/10.1016/j.artmed.2022.102276>
36. Nitefor, L.: Clinical features of immunohistochemical diagnosis of sarcoma and pholid tumors of the breast. *APP* (2017). <https://doi.org/10.11603/24116-4944.2017.2.8036>
37. Olatunji, S., Alotaibi, S., Almutairi, E.: Early diagnosis of thyroid cancer diseases using computational intelligence techniques: a case study of a Saudi Arabian dataset. *Comput. Biol. Med.* **131**, 104267 (2021). <https://doi.org/10.1016/j.compbiomed.2021.104267>
38. de Oliveira Silva, L., Barros, A., Lopes, M.: Detecting masses in dense breast using independent component analysis. *Artif. Intell. Med.* **80**, 29–38 (2017)
39. Peng, Y., Yao, B., Jiang, J.: Knowledge-discovery incorporated evolutionary search for microcalcification detection in breast cancer diagnosis. *Artif. Intell. Med.* **37**(1), 43–53 (2006). <https://doi.org/10.1016/j.artmed.2005.09.001>
40. Pham, B., Gaonkar, B., Whitehead, W.: Cell counting and segmentation of immunohistochemical images in the spinal cord: comparing deep learning and traditional approaches. In: 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 842–845. IEEE, Honolulu, HI (2018)
41. Pouliakis, A., Karakitsou, E., Margari, N.: Artificial neural networks as decision support tools in cytopathology: past, present, and future. *Biomed. Eng. Comput. Biol.* (2016). <https://doi.org/10.4137/BECB.S31601>
42. Prasad, R., Udupa, K.: BChisto-Net: breast histopathological image classification by global and local feature aggregation. *Artif. Intell. Med.* **121**, 102191 (2021). <https://doi.org/10.1016/j.artmed.2021.102191>
43. Ramos-Vara, J.: Technical aspects of immunohistochemistry. *Vet. Pathol.* **42**, 405–426 (2005). <https://doi.org/10.1354/vp.42-4-405>
44. Salimans, T.: Improved techniques for training GAN. Improved techniques for training GAN. <http://arxiv.org/abs/1606.03498> (2021)
45. Xiong, H., Sharan, R.: Weak label based Bayesian U-Net for optic disc segmentation in fundus images. *Artif. Intell. Med.* **126**, 102261 (2022). <https://doi.org/10.1016/j.artmed.2022.102261>
46. Yager, J., Davidson, N.: Estrogen carcinogenesis in breast cancer. *N. Engl. J. Med.* **354**, 270–282 (2006). <https://doi.org/10.1056/NEJMra050776>
47. Zhang, C., Gu, J.: AI in spotting high-risk characteristics of medical imaging and molecular pathology. *Precis. Clin. Med.* **4**, 271–286 (2021). <https://doi.org/10.1093/pcmedi/pbab026>
48. Zhang, H., Goodfellow, I., Metaxas, D.: Self-attention generative adversarial networks (2019)



Approaches and Techniques to Improve Machine Learning Performance in Distributed Transducer Networks

Mykola Hodovychenko¹ (✉) , Svitlana Antoshchuk¹ , Ivan Lobachev² ,
Thorsten Schöler³ , and Mykhaylo Lobachev¹ 

¹ Odessa Polytechnic National University, Odessa, Ukraine
nick.godov@gmail.com

² Intel Corporation Hillsboro, Oregon, USA

³ University of Applied Sciences, Augsburg, Germany
thorsten.schoeler@hs-augsburg.de

Abstract. Purpose of the study is to improve efficiency of distributed transducer networks, which use models and methods of machine learning to analyze measures from sensors and perform reasoning on this measured data. The proposed model makes it possible to process data from different types of sensors to solve classification and regression problems. The model has hierarchical structure consisting of three-level calculators, which allows it to be used in the edge computing approach and to process data from different types of sensors, with varying measures dimensions and frequency of new data. To form the input tensor of a hierarchical neural network model, an algorithm for pre-processing and formatting input data from measures of different sensor types is proposed, considering the spatial and temporal characteristics of the obtained measurements. Also a method of neural networks compression based on hidden layers neurons pruning is proposed. Suggested method implements a unified approach to the compression of convolutional, recurrent and fully connected neural networks to solve classification and regression problems. The core mechanism of the proposed method is based on the dropout operation, which is used as a technique of neural network regularization. The method estimates rational probability of hidden layers neurons dropout on the basis of the neuron excessiveness parameter, which is estimated with the help of a special “trimmer” network. The proposed techniques could reduce time, memory and energy consumption for the neural network inference.

Keywords: Transducer networks · Internet of Things · Machine learning · Neural network compression · Multisensory data

1 Introduction

Recent advances in sensors, wireless technology and embedded processors have led to the development of low-power and low-cost devices that can be grouped in

a network and connected to the Internet. These are key components of the new Internet of Things (IoT) paradigm. IoT devices use the Internet as an infrastructure to connect each other in order to communicate, coordinate and collaborate to provide advanced control and monitoring services [16]. IoT systems covers different areas of human life, such as health monitoring, smart home, smart cities, smart industry, etc. [10].

IoT devices interact with the physical world through a distributed transducer network (network of sensors and actuators) to maintain the required parameters [19]. The gateway nodes connects to the local IoT network via the Internet and aggregates the collected data and even offers processing services. Cloud servers or cloud applications provide analysis or storage services.

A key element of most IoT systems is the intelligent learning mechanism for forecasting (regression, classification, and clustering), data mining and pattern recognition [12]. Among many machine learning techniques, deep learning (DL) is increasingly used in many IoT systems. DL models generally brings two important improvements over traditional machine learning approaches at the learning and inference stages. First, they reduce the need to manually create and refine data features. Thus, some features that may not be obvious to the human eye can be easily refined by deep learning models. In addition, DL models improves inference accuracy [9].

Despite many advantages that brings the use of mathematical apparatus of deep learning, their practical application requires proper adaptation, taking into account the architecture and characteristic of distributed transducer network.

Existing deep learning methods are not suitable for most IoT system (such systems are often consists of huge number of distributed heterogeneous devices that produce a constant data flow [9]). In addition, IoT devices often have limited computing and power resources. Considering this, deep learning models must be adapted to solve specific problems of sensory data analysis. One of such technique is reducing the redundancy of the model structure. This will allows to use computing and energy resources more efficiently.

The aim of the research is to increase the efficiency of distributed transducer networks based on machine learning by developing neural network models and methods of sensory data analysis.

The main contributions of this paper are summarized as follows. The paper investigates approaches to improve the efficiency of machine learning models that are used in distributed transducer networks to analyze data from network sensors. The model uses convolutional and recurrent neural networks, which allows to organize computations on a hierarchical principle without the use of a centralized cloud server.

It is shown that the functioning of modern IoT systems is based on the analysis of the measurements of many sensors of different types, with different data dimensions and the frequency of obtaining new data. To use deep neural networks as a tool for analysis and processing of sensor measurements, it is necessary to solve the problem of pre-processing of incoming data and converting them into the desired format for further use as input for neural networks of

different architecture. To solve this problem, an algorithm of sensor readings pre-processing and formatting is proposed.

The structures of typical neural network models can be excessive to solve various problems of machine learning. The redundancy of typical neural network models leads to the need of more powerful computing nodes and energy consumption without increasing efficiency. Refining the structures of neural networks to solve specific problems of sensory data analysis will allow the use of computing and energy resources more efficiently. To solve this problem, a method of neural network compression is proposed in order to reduce the requirements for computing and energy resources by reducing the number of connections between the neurons of the network.

The rest of the paper is structured in the following way. Section 2 reviews recent publications correlated with the challenges of machine learning methods usage in distributed transducer networks. Section 3 describes the proposed hierarchical neural network model, data fusion algorithm and neural network compression method. Section 4 describes conducted experimental studies to confirm the effectiveness of the proposed approaches. Section 5 describes the results of carried out experiments. Section “Conclusions” summarizes the research results and indicates the direction for future work.

2 Literature Review

Literature review of the scientific publications related to the Internet of Things area has established several important approaches for improving the efficiency of machine learning performance in distributed transducer networks.

Data Analysis Infrastructure. An important factor in ensuring the efficiency of distributed transducer networks is the infrastructure of collection and analysis of input sensor data [2]. Cloud computing is one of the most common infrastructure for data analysis in many modern IoT systems. For example, resource-intensive tasks such as video analysis require large and complex models with a vast number of computing resources. Thus, the design of scalable and high-performance cloud-based models and algorithms for deep neural networks that can perform analysis of large data sets still remains an important task. Centralized IoT systems with cloud computing infrastructure assumes storing and processing data away from end users and data sources. This approach offers the advantages of huge computing power, high scalability and reliability, as well as low initial cost of such systems. However, with zettabyte-sized traffic and the rapid growth of IoT systems, transferring all data to the cloud for processing is impractical or even impossible [1].

Edge Computing approach emerged as a way to solve these problems by moving computing to the boundaries of the network and data sources. The key idea of edge computing is to reduce network traffic by bringing computations closer to data sources [18]. Examples of the use of edge computing includes autonomous vehicles, medical devices, retail advertising, smart speakers and video conferencing. Also, edge computing has been studied in detail in mobile

computing, where data is stored at the edges to reduce latency, improve user performance, reduce battery consumption, and introduce location awareness. The edge computing architecture is particularly suitable for applications with ultra-low latency requirements, as well as for content delivery and caching [6].

Data Collection and Aggregation Model. In addition to choosing the infrastructure of data analysis, to improve the efficiency of distributed transducer networks, it is necessary to address the issue of data acquisition, determination of their properties and characteristics. The data generated by IoT systems differs from the general big data. To better understand the requirements for IoT data analysis, it is necessary to explore the properties of IoT data and how it differs from general big data [3]. IoT data have the following characteristics:

- large-scale streaming data: a large number of sensors in the distributed transducer network constantly collect data. This leads to a huge amount of streaming data;
- heterogeneity: different sensors in IoT systems measures different physical values, which leads to heterogeneity of data;
- correlation of data in time and space: in most IoT systems sensor devices are located at a certain place, and thus have a location and timestamp characteristics for each of the data elements;
- high noise data: due to the small fragments of data collected by sensors, many of these data in IoT systems may have errors and noise during reception and transmission.

Data generated by the sensors of IoT systems can be transmitted in the stream mode, or accumulated as a source of big data. Knowledge can be gained from big data in a few days after direct data generation, while knowledge from streaming data should be obtained in the range of a few hundred milliseconds to a few seconds.

Data aggregation is also critical factor of machine learning methods performance. Data in IoT systems is created by different sensors. Data merging and sharing is the key to developing effective modern IoT systems. Effective merging of data of different types and forms leads to improved data quality and machine learning models inference. Data merging is defined as “the theory, methods and tools used to combine sensory data or data derived from sensory data into a common representative format” [15]. Timely integration and analysis of data obtained from sensors of IoT systems and other sources, which allows for highly efficient, reliable and accurate decision-making is one of the challenges facing developers of modern IoT systems.

Adaptation of Machine Learning Models. Before the advent of the IoT, most research on deep learning focused on improving models and algorithms for efficient big data processing by deploying deep learning models on cloud platforms. The emerging of IoT systems has opened up a completely different area of research, with the scale of the problem shrinking to devices with limited resources and the need for real-time data analysis. At present, deep neural networks are virtually impossible to train on edge devices with limited resources, because deep

learning methods require a large number of resources, energy and memory. In most cases, the available resources are insufficient to deploy pre-trained neural networks to perform inference [13].

Recent studies have shown that many parameters stored in deep neural networks may be redundant. Thus, effective removal of parameters or layers can significantly reduce the complexity of DL models without significantly reducing its efficiency [5]. The neural network complexity reduction approach is called neural network compression. Compressed neural networks need less time for training and inference, and have less storage capacity and computing power requirements. Neural network compression methods can be divided into five categories, depending on the approach to the compression process: network pruning, sparse representation, bit accuracy, knowledge distillation, and other methods that cannot be categorized [14].

Based on a review of ways to improve the efficiency of distributed transducer networks, this paper proposes a neural network model for the analysis of multisensory data, which allows to adapt machine learning technologies to work in the conditions of edge computing. Also, in order to reduce the requirements for computing and energy resources, a method of neural network compression based on the approach of network pruning is proposed.

3 Materials and Methods

Hierarchical Neural Network Model. The proposed hierarchical model integrates convolutional neural networks and recurrent neural networks. First, the input from the sensors is formatted and divided into time intervals for processing. The data of each time interval of an individual sensor is first fed to the convolutional neural network, where the encoding of relevant local features in the data stream from the sensor is conducted. Then from the outputs of individual convolutional neural networks, the encoded data is fed to a single convolutional fusion network, which simulates the relationship between multiple sensors. Next, a recurrent neural network is used to extract time patterns from these data. Next, an affine transformation is used for the estimation problem, and a transformation with normalized exponential function is used for the classification problem.

This model solves the problem of joint processing of multisensory data for evaluation or classification tasks using time data. For evaluation tasks, the model tries to obtain output from sensor measurements with noise patterns. In this case, the neural network model acts as an approximating function. For classification tasks, the neural network acts as a feature extractor that uses information from spatial and temporal domains.

The model structure is presented in Fig. 1 and comprises of three level calculators.

First level calculator is on a low-power edge node, which take sensor measurements or provide instruction to the actuators of the distributed transducer network. Each 1st level calculator processes the readings of a distinct sensor and is a simple convolutional network with three convolutional layers. The input



Fig. 1. The structure of hierarchical neural network model

data for this calculator is a matrix X^m , where $m \in \{1, \dots, M\}$, M - number of sensors. The task of the first-level calculator is to elicit relations in the spatial and frequency domain from the sensor measurements $X_t^{(m)}$. Nearby frequencies often has a lot of patterns in frequency domain. To elicit this patterns, we could use two-dimensional filter to $X_t^{(m)}$ with the form $(dim^{(m)}, c1)$ to produce output $X_t^{(m,1)}$. After that procedure, we could sequentially apply filters with sizes $(1, c2)$ and $(1, c3)$ to get $X_t^{(m,2)}$ and $X_t^{(m,3)}$ respectively. After that, the resulting matrix is flattened into the vector $x_t^{(m,3)}$, which is transferred to the 2nd level calculator.

Second-level calculator deployed on a medium-powered edge node or on a gateway node. Each 2nd level calculator is represented by a distinct convolutional fusion network, which processes the results of a set of 1st level calculators to find the relationships between the data of individual sensors. Output vectors from the 1st level calculators formed into the matrix $X_t^{(3)}$ and fed to the input of 2nd level calculator. A filter with the size $(M, c4)$ is applied to the input matrix to obtain relationships between M sensors. Then, filters $(1, c5)$ and $(1, c6)$ is successively applied to the matrix $X_t^{(4)}$ to obtain $X_t^{(5)}$ and $X_t^{(6)}$ respectively. Resulting matrix is then flattened into the vector $x_t^{(f)}$, the vector concatenates with the time intervals $|\tau|$ to get vector $X_t^{(rec)}$, which is transferred to the 3rd level calculator.

Third-level calculator deployed on a gateway node and carries intensive computing. This calculator has a recurrent neural network in order to approximate functions and extract features from time sequences. It is proposed to use Gated

Recurrent Unit (GRU) as recurrent network model in order to reduce computations in 3rd level calculator. This model shows efficiency comparable to Long short-term memory (LSTM) model but has a lower complexity. Inputs $\{x_t^{(rec)}\}$ from the 2nd level calculators are fed to the input of recurrent network. Outputs $\{x_t^{(out)}\}$ are then fed the output layer. The structure of the output layer differs, depending on the task type (classification or regression). For the regression task, $x_t^{(out)}$ encodes the physical values of the time frame t , each element of $x_t^{(out)}$ is within range ± 1 . Thus, the output layer will be fully connected layer on top of separate time frame with global parameters W_{out} and b_{out} . For the classification task, vector $x_t^{(out)}$ is a feature vector on the time frame t . Input vectors $\{x_t^{(out)}\}$ should be converted into feature vector with a fixed length. Then, features could be averaged over time and fed to the softmax layer in order to obtain probability of the category \hat{y} .

Formatting Sensor Data. According to the proposed hierarchical model, the measurements from sensors should be shaped into a tensor X^m , which is then fed to the input of convolutional layer of the 1st level calculator. The sensors of distributed transducer network measures wide range of physical values and produces data in different shapes and forms. In order to form the input matrix X^m it is necessary to process and format the sensor measurements while keeping information in spatio-temporal domain. To solve this problem, we propose an algorithm for forming an input tensor from a set of measurements that is gathered from sensors of various types over time.

Suppose, we have M different sensors $Sensor = \{Sensor_m\}$, where $m \in \{1, \dots, M\}$. The algorithm has following steps:

1. pick the sensor $Sensor_m$, which produces data over time;
2. compose the matrix O with shape $dim^{(m)} \times n^{(m)}$, where $dim^{(m)}$ is the dimension of $Sensor_m$ measurement and $n^{(m)}$ is the number of measurements;
3. form a vector p of time stamps with the shape $n^{(m)}$;
4. decompose the matrix O and vector p into the set $Q = \{(O_t^{(m)}, p_t^{(m)})\}$ of time frames with the fixed width τ , where $|Q| = T$;
5. for each element of Q the Fourier transform is applied;
6. form a tensor $X^{(m)}$ from the results of a Fourier transform. Formed tensor will have a shape $dim^{(m)} \times 2f \times T$, where f is the dimension of the frequency domain and contains a pair of amplitude and phase.

Neural Network Compression. The base operation of the proposed compression method is a method of regularization of deep neural networks, known as dropout. The dropout operation corresponds to each neuron the probability of its exclusion $p^{(lr)}$, where $lr = 1, \dots, L$ and L is the ordinal number of network layer. The main parameter of this operation is the probability of exclusion, which must be evaluated in such a way to create a rational network structure that maintains the accuracy of inference and minimizes resource usage [11].

To calculate the exclusion probability parameter, the proposed method uses the information contained in the network itself. To calculate the exclusion probability parameter, the proposed method uses the information contained in the

network itself. To achieve this goal, the redundancy parameter of the hidden layer neuron is introduced in the method. During neural network compression, a neuron with a higher redundancy score has a higher chance of being excluded. A feature of this method is the use of a special neural network that plays the role of a trimmer (hereinafter referred to as the trimmer network), which takes the weights of the neurons of each layer as input, calculates the redundancy parameter and obtains the value of the exclusion probability for each neuron of this layer. The proposed method can be described as a sequence of the following steps:

1. exclusion operation is applied to the neurons of hidden layer. This operation drops neurons with the initial probability $p^{(l)}$;
2. the trimmer network $trm \sim \mu_\xi(W)$ is initialized. Matrices $W^{(l)}$, formed from the weights of hidden layers and fed to the trimmer network sequentially. Trimmer network calculates redundancy values for each element of the layer and then evaluates dropout probabilities;
3. iterative process of training of the trimmer network $trm \sim \mu_\xi(W)$ and the source network $F_W(x|z)$ is conducted. The trimmer network is trained to obtain more accurate dropout probabilities, which leads to more efficient output network compression.

The compression output values are dropout mask \hat{e} and a set W of fine-tuned weights of the source neural network.

The basic idea of proposed compression approach is that neural network with exclusion operation could be described as Bayesian neural network with Bernoulli variational distribution [7]:

$$\begin{aligned} z_{[j]}^l &\sim \text{Bernoulli}(p_{[j]}^{(l)}) \\ \tilde{W}^{(l)} &= W^{(l)} \text{diag}(z^{(l)}), \\ Y^{(l)} &= X^{(l)} \tilde{W}^{(l)} + b^{(l)}, \\ X^{(l+1)} &= f(Y^l), \end{aligned} \tag{1}$$

where $l = 1, \dots, L$ is the layer number, $W^{(l)}$ is the weight matrix for layer l , b^l is the bias vector, $X^{(l)}$ is the input tensor and $f(\cdot)$ is the activation function.

Source neural network is defined as $F_W(x|z)$. It gets x as network input and produces prediction based on dropout z and parameters W . It is assumed that $F_W(x|z)$ is already trained. Trimmer network is defined as $trm \sim \mu_\xi(W)$. It takes the weights of source neural network and generates the probability distribution of the mask vector z based on its own parameters ϕ . In order to enhance trimmer network to exclude neurons of hidden layers of the source network, proposed method uses the following function

$$L = E_{Z \sim \mu_\phi} [L(y, F_w(z))] = \sum_{z \sim \{0,1\}^{|z|}} \mu_\phi(W) \cdot L(y, F_W(x|z)) \tag{2}$$

where $L(\cdot, \cdot)$ is the objective function of the source network. This function is the expected loss of the network over the dropout operation. Proposed method

enhanced trimmer network and source network in a step-by-step manner. It tries to reduce loss function value as described in formula above by using gradient descent.

4 Experiment

An experimental study of the proposed hierarchical neural network model and compression method was performed for the task of recognizing human actions based on the readings of motion sensors, accelerometer and gyroscope. The dataset presented in [17] is used as a data set. This dataset contains accelerometer and gyroscope readings obtained from smartphones and smartwatches of people who performed certain actions at random. The data set contains the readings of 9 people who performed 6 actions (cycling, sitting, standing, walking, climbing stairs, descending stairs). Data were obtained from 6 types of mobile devices.

The proposed hierarchical neural network model was used as a neural network to solve this task. The original and compressed network models characteristics and performance are shown in Table 1.

Table 1. Source network and compressed network characteristics

Layer		Source network				Compressed network			
		Neurons	Neurons	Parameters	Parameters	Neurons	Left parameters, %		
cnv1A	cnv1B	64	64	1200	1200	19	18	29.68%	28.13%
cnv2A	cnv2B	64	64	12400	12400	18	15	9.47%	6.44%
cnv3A	cnv3B	64	64	12400	12400	22	22	11.19%	7.82%
cnv4(2 × 8)		64		65500		11		5.58%	
cnv5(1 × 6)		64		24600		13		3.14%	
cnv6(1 × 4)		64		16400		16		4.58%	
rnn1		120		227500		26		5.4%	
rnn2		120		86400		30		6.18%	
fulcon1		6		720		6		25.94%	
Total				477120				6.14%	
Accuracy		94.6%				94.6%			

MagBase, MagBase-R and SFAC methods were used for comparison with the proposed compression method.

MagBase method, based on the neural network pruning algorithm using magnitude-based pruning modules [8]. This method is based on the pruning of weights in convolutional cores and fully connected layers based on the parameter of the connection magnitude. The method retrains the network after each pruning and has the ability to recover deleted neurons. For convolutional and fully connected layers, the optimal pruning threshold is calculated separately.

MagBase-R method is a modification of the parent method, which allows it to be applied to recurrent networks.

The SFAC method is based on the method of sparse coding and factorization. The method simplifies fully connected layers by finding the optimal code vectors and layer coding based on the sparse coding method. The algorithm tries to compress the convoluted layers using the matrix decomposition method. The search for optimal code vectors is carried out using greedy algorithm [4].

The structure and performance of the proposed compression method and analogous methods are presented in Table 2. The hierarchical network contains both convolutional and recurrent layers. Because the SFAC and MagBase methods do not apply directly to recurrent layers, the recurrent layers remain intact during compression. The proposed compression method provides the highest degree of compression of the source network (based on the number of parameters remaining after compression) without reducing the accuracy of model inference.

Table 2. Comparison of neural network compression by the proposed method and analogue methods

Layer		Proposed method		MagBase-R		MagBase		SFAC	
		Left params, %		Left params, %		Left params, %		Left params, %	
cnv1A	cnv1B	29.68%	28.13%	91.8%	95.6%	50.4%	59.83%	100%	100%
cnv2A	cnv2B	9.47%	6.44%	70.2%	77.6%	25.2%	40.7%	113%	113%
cnv3A	cnv3B	11.19%	7.82%	69.8%	66.1%	32.2%	35.4%	113%	113%
cnv4(2 × 8)		5.58%		40.2%		20.1%		53.5%	
cnv5(1 × 6)		3.14%		27.1%		18.2%		100%	
cnv6(1 × 4)		4.58%		24.2%		11.7%		100%	
rnn1		5.4%		1.1%		100%		100%	
rnn2		6.18%		3.3%		100%		100%	
fulcon1		25.94%		98.4%		98.9%		72%	
Total		6.14%		16.8%		73.9%		95.1%	
Accuracy		94.6%		94.6%		94.6%		93.8%	

5 Results and Discussion

Figure 2 shows a compromise between model accuracy and memory consumption. Due to the fact that the proposed compression method provides a more compact matrix representation of compressed layers, memory consumption was lower than that of analogous methods.

The compromise between the inference time and the accuracy of the compressed models is shown in Fig. 3. The model compressed by the proposed method provides output in 36.7 ms, which is 71.4% faster than the original model. The MagBase and MagBase-R methods allow you to get a model that runs faster

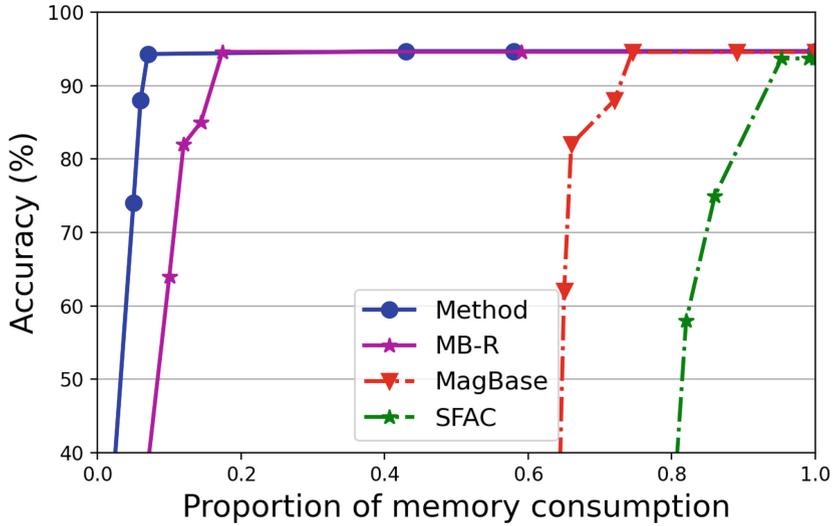


Fig. 2. Compromise between accuracy and memory consumption

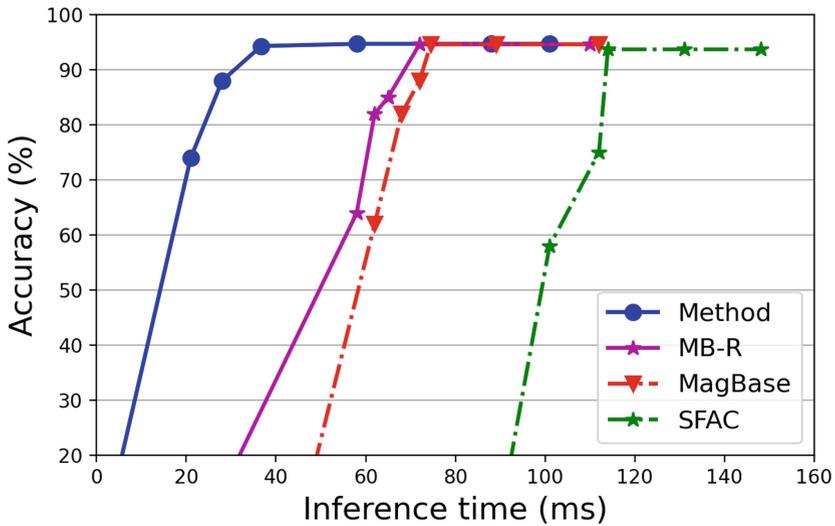


Fig. 3. Compromise between accuracy and inference time

than the model obtained by the SFAC method. The original model contains one-dimensional convolutional cores, which led to the fact that the method of matrix decomposition used in SFAC did not reduce the number of parameters, but only increased the redundancy and the number of model parameters. Therefore, the SFAC method has limitations on the compressed network structure.

The compromise between model accuracy and energy consumption is shown in Fig. 4. The proposed method reduced the energy consumption per model output by 71.9% compared to the original model, which turned out to be the best indicator among the compared methods.

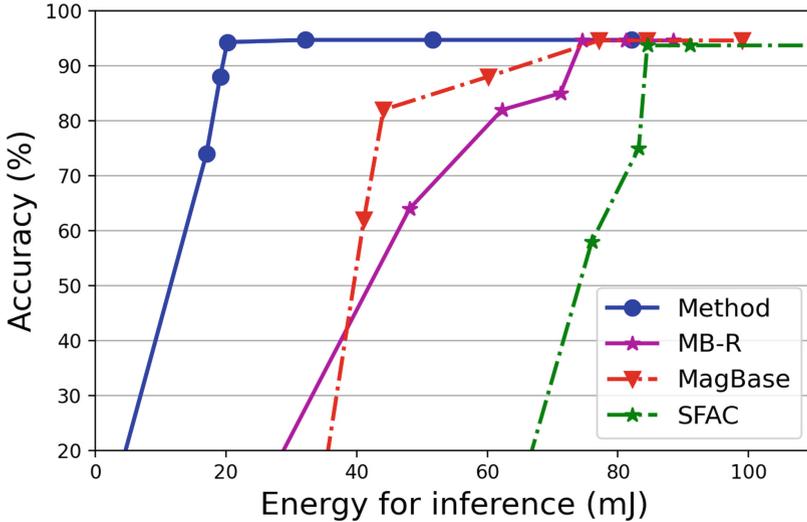


Fig. 4. Compromise between accuracy and energy for inference

6 Conclusions

This paper presents a hierarchical neural network model for multisensory data analysis, which allows you to organize neural network calculations on a hierarchical basis without the use of a centralized cloud server. The model uses convolutional and recurrent neural networks and allows to solve tasks of classification, identification and regression by processing multisensory data received over a period of time. Also, in order to adapt machine learning models for their use in modern distributed transducer networks, a method of neural network compression is proposed to reduce the requirements for computing and energy resources by pruning connections between network neurons. In the process of compression, the elements of the hidden layers can be excluded based on the probability parameter, which allows you to get a network structure with fewer elements. The probability of exclusion should be selected in such a way as to generate a network structure that maintains the accuracy of the inference and minimizes resource consumption. To obtain the probability of exclusion of the neuron, the redundancy parameter of the hidden layer neuron is presented. In terms of the compression process of the neural network model, an element with a higher redundancy rate is more likely to be excluded. Experiments for the problem of classifications of human activities shows that the compression of the

proposed hierarchical neural network model of has reduced network inference time by 71.4% and power consumption by 71.9% in compared to the source model. Possible directions for the continuation of the work are the further study of dependencies between the structure of the neural network and the efficiency of its operation, which will further reduce the inference time and power consumption.

References

1. Assunção, M.D., Calheiros, R.N., Bianchi, S., Netto, M.A., Buyya, R.: Big data computing and clouds: trends and future directions. *J. Parallel Distrib. Comput.* **79–80**, 3–15 (2015). <https://doi.org/10.1016/j.jpdc.2014.08.003>
2. Athmaja, S., Hanumanthappa, M., Kavitha, V.: A survey of machine learning algorithms for big data analytics. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pp. 1–4 (2017). <https://doi.org/10.1109/ICIIECS.2017.8276028>
3. Aziz, F., Chalup, S.K., Juniper, J.: Big data in IoT systems. In: Khan, J.Y., Yuce, M.R. (eds.) *Internet of Things (IoT): Systems and Applications*, chap. 2. Pan Stanford Publishing Pte. Ltd., Singapore (2019)
4. Bhattacharya, S., Lane, N.D.: Sparsification and separation of deep learning layers for constrained resource inference on wearables. In: *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, SenSys 2016*, pp. 176–189. Association for Computing Machinery, New York (2016). <https://doi.org/10.1145/2994551.2994564>
5. Cheng, Y., Yu, F.X., Feris, R.S., Kumar, S., Choudhary, A., Chang, S.F.: An exploration of parameter redundancy in deep networks with circulant projections (2015). <https://doi.org/10.48550/ARXIV.1502.03436>. <https://arxiv.org/abs/1502.03436>
6. El-Sayed, H., et al.: Edge of things: the big picture on the integration of edge, IoT and the cloud in a distributed computing environment. *IEEE Access* **6**, 1706–1717 (2018). <https://doi.org/10.1109/ACCESS.2017.2780087>
7. Gal, Y., Ghahramani, Z.: Bayesian convolutional neural networks with Bernoulli approximate variational inference (2015). <https://doi.org/10.48550/ARXIV.1506.02158>. <https://arxiv.org/abs/1506.02158>
8. Guo, Y., Yao, A., Chen, Y.: Dynamic network surgery for efficient DNNs (2016). <https://doi.org/10.48550/ARXIV.1608.04493>. <https://arxiv.org/abs/1608.04493>
9. Lin, T.: Deep learning for IoT (2021). <https://doi.org/10.48550/ARXIV.2104.05569>. <https://arxiv.org/abs/2104.05569>
10. Lobachev, I., Antoshchuk, S., Hodovychenko, M.: Distributed deep learning framework for smart building transducer network. *Appl. Aspects Inf. Technol.* **4(2)**, 127–139 (2021). <https://doi.org/10.15276/aaif.02.2021.1>
11. Lobachev, I., Antoshchuk, S., Hodovychenko, M.: Methodology of neural network compression for multi-sensor transducer network models based on edge computing principles. *Herald Adv. Inf. Technol.* **4(3)**, 232–243 (2021). <https://doi.org/10.15276/haif.03.2021.3>
12. Mahdavinjad, M.S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth, A.P.: Machine learning for internet of things data analysis: a survey. *Digit. Commun. Netw.* **4(3)**, 161–175 (2018). <https://doi.org/10.1016/j.dcan.2017.10.002>

13. Marco, V.S., Taylor, B., Wang, Z., Elkhatib, Y.: Optimizing deep learning inference on embedded systems through adaptive model selection (2019). <https://doi.org/10.48550/ARXIV.1911.04946>. <https://arxiv.org/abs/1911.04946>
14. Mishra, R., Gupta, H.P., Dutta, T.: A survey on deep neural network compression: challenges, overview, and solutions (2020). <https://doi.org/10.48550/ARXIV.2010.03954>. <https://arxiv.org/abs/2010.03954>
15. Ranjan, R., et al.: City data fusion: sensor data fusion in the internet of things. *Int. J. Distrib. Syst. Technol.* **7**(1), 15–36 (2016). <https://doi.org/10.4018/IJDST.2016010102>
16. Samie, F., Bauer, L., Henkel, J.: IoT technologies for embedded computing: a survey. In: *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES 2016*. Association for Computing Machinery, New York (2016). <https://doi.org/10.1145/2968456.2974004>
17. Stisen, A., et al.: Smart devices are different: assessing and mitigating mobile sensing heterogeneities for activity recognition. In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015*, pp. 127–140. Association for Computing Machinery, New York (2015). <https://doi.org/10.1145/2809695.2809718>
18. Sulieman, N.A., Ricciardi Celsi, L., Li, W., Zomaya, A., Villari, M.: Edge-oriented computing: a survey on research and use cases. *Energies* **15**(2) (2022). <https://doi.org/10.3390/en15020452>. <https://www.mdpi.com/1996-1073/15/2/452>
19. Xia, F., Tian, Y.C., Li, Y., Sun, Y.: Wireless sensor/actuator network design for mobile control applications (2008). <https://doi.org/10.48550/ARXIV.0806.1569>. <https://arxiv.org/abs/0806.1569>



Investigation of the Impact of Primary Data Processing on the Results of Neural Network Training for Satellite Imagery Recognition

Dmytro Soldatenko^(✉)  and Viktoriia Hnatushenko 

Ukrainian State University of Science and Technologies, Dnipro, Ukraine
zeirison@gmail.com

Abstract. Research and recognition of high-resolution multi-spectral satellite images is a complex and vital task in modern science. Of particular difficulty is the recognition of satellite images without additional channels, based only on RGB, in addition to very homogeneous classes with similar primary features. This article discusses the creation and configuration of a neural network architecture based on ConvNet with subsequent training. High-resolution satellite images from the Landsat 8-9 OLI/TIRS C2 L2 without additional channels to create datasets necessary for the neural network to complete the task. The paper presents four experiments on ranking the input data of the neural network to identify their influence on the final result of recognition, regardless of the settings and architecture of the neural network itself. Pre-processing of the input data was based on annotating the images of each class and then creating masks for them, namely for the classes: water, trees, and fields. In experiments with prepared validation input data, the increase in class recognition was up to 54.44%. Conclusions have about each experiment and the influence of input data on the result of satellite image processing.

Keywords: Neural network · Image recognition · Data processing · Satellite imagery · Artificial intelligence

1 Introduction

The problem of recognizing and classifying satellite images from a minimum set of channels, namely, in our case, only RGB [3], is an urgent and rather time-consuming task. The task requires meticulous preliminary preparation of the input dataset and the construction of suitable neural network architecture. Of particular difficulty are images with relatively homogeneous classes [9], such as greenery, fields, forests [17], marshy water bodies [14], or just water, as an example from the research in Fig. 1.

Usually, the listed classes have a somewhat similar color consistency and, in some cases, also contain pretty similar features. The absence of additional



Fig. 1. Example of classes images from research: a) water; b) trees; c) fields

channels in the image does not allow the neural network to approach self-training more carefully. We have to mainly focus on an additive color model consisting of 3 channels.

For each task associated with training a neural network to recognize satellite images, it is necessary to prepare the input datasets for each class that needs to be recognized, and it can be hundreds of images. In most cases, the preparation of primary materials is manual work, which is a challenging task, and we need to know how much input data is necessary for each class. In some cases, the preparation of raw materials can be done automatically, for example, using an image pre-processing program. For example, there are many ready-made solutions for image annotation [22], which reduces the time for preparing data and creating masks.

It is worth noting that many problems in recognition can be solved by increasing the amount of input data, such as the number of input images and their channels. However, these solutions lead to an increase in labor costs and preparation time, which is not always possible. This fact indicates the relevance of the problem. In this research, we will provide information on building a neural network architecture for working with a small sample of data and conduct experiments to find the least labor-intensive and high-quality option with the necessary amount of training and validation data.

2 Literature Review

As noted in the Introduction section, the direction of this research, namely the recognition of satellite images, is relevant and is being tried on in various fields and scientific research. Researchers who face problems when working in this industry are devoted to many scientific papers.

The research [19] presents the results of research on attempts to improve the architecture of a neural network based on ConvNet to improve the accuracy of recognition and classification. The authors of this paper investigate the effect of the depth of a convolutional network on its accuracy in setting up large-scale image recognition. They thoroughly evaluate networks with increasing depth using an architecture with tiny (3×3) convolutional filters, which increases recognition by increasing the depth to 16–19 layers. They also consider an analysis of using a smaller receptive window and a more minor step of the first convolutional layer. This research is quite suitable for reconciliation when working with a small number of input data, given the use of tiny convolutional filters.

The study [24] discusses the use of their neural network architecture called Cross-Layer Neurons Networks (CLNN). CLNN allows us to train efficient deeper neural networks in connection with interlayer neurons for the synthesis of information (functions) received from all lower-level layers and then sent to higher-level layers. This option showed quite good results, but it does not allow obtaining optimal accuracy with small amounts of input data and requires more time for training.

Improving the quality of image recognition based on the calculation of spatial clues that are resistant to image rotation is presented in the study [2]. The authors proposed the calculation of spatial clues by calculating the histograms of orthogonal vectors by calculating the magnitude of orthogonal vectors between pairs of identical visual words (PIVW) relative to the geometric center of the image. The proposed approach provides discriminatory features and invariance to geometric changes in images, which leads to an increase in classification accuracy.

Despite all the achievements in this subject area, the problem of requiring a large amount of input data when training neural networks to recognize even a minimal set of classes has not been solved nowadays. One of the solutions to this problem can be achieved based on selecting the optimal minimum quantity and quality of primary data for various classes, as well as the degree of their processing and the correct construction of the neural network architecture, which is partially used in various studies [4, 21, 23].

The objective of the research is to compare the influence of the quality of input data, creating a suitable neural network architecture based on CNN and search for the most optimal, less time-consuming option and with minimum loss in the quality of classification and recognition.

3 Problem Statement

When working with satellite images, the recognition and classification of objects is a significant, fundamental, and complex problem since each class and object has its parameters and not always unique features. Even more challenging is the paucity of input data, as in our case, a small number of images for training, 20 for each class, and a small number of channels in them, only RGB. It also requires the correct selection and construction of the neural network's architecture, which will allow it to cope with the task.

Thus, our research aims to improve the accuracy of classification and recognition of a high-resolution multispectral satellite image based on preprocessing methods and preparation of primary data using a neural network. Achieving this goal requires the additional design of the neural network architecture [20] based on ConvNet.

4 Materials and Methods

The initial approach to processing and analyzing multi-spectral satellite images [5] was spectral analysis with pre-processing by Gaussian filters [10].

However, the results showed an average accuracy of 57–62%. Also, applying the previously mentioned method is quite labor-intensive and did not bring the desired result, so we adopted a different approach in favor of other algorithms and solutions. Subsequently, variants were undertaken with an increase, in contrast, to increase the definition of the clarity of the boundaries and further classify individual parts of the obtained images by color. However, this option did not bring the desired result since the application of this method does not consider the blurring of the boundaries, which also did not allow obtaining the desired result.

Since using linear algorithms did not achieve the required results and obtained the necessary accuracy, we chose deep learning, namely the use of a convolutional neural network. Also, the choice of a convolutional neural network was because their use and training allow us to get a more accurate result since they have aimed at the most accurate and efficient pattern recognition.

4.1 Convolutions

A convolutional neural network [1] works based on filters that recognize the specified characteristics of the input signals or images. The filter is one or more scale matrices adjusted to the required in the learning process [6]. The filter passes through the image and determines the presence of the required parameters, for example, a curved border in a specific area of the image.

The main operation is the convolution operation and is the sum of the products of the scales [25] represented in the filter or its elements and the matrix of input signals, namely the image. The convolution operation determines by the formula (1). The operation is performed on a pair of matrices A of dimension $n_x \times n_y$ and B of dimension $m_x \times m_y$. The result is a matrix $C = A * B$ of dimension $(n_x - m_x + 1) \times (n_y - m_y + 1)$.

$$C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=1}^{m_y-1} A_{i+u,j+v} B_{u,v} \quad (1)$$

If the characteristic is appropriately present in a given area of the image, the result of the operation will be a relatively large number. Otherwise, the resulting number will be small.

Because of moving the filter along the image, a matrix of the elements will result from a single convolution.

Note that the number of filter elements must correspond precisely to the number of color channels of the original image. In this case, the result of the operation will be the correct number. For example, the image is color in our case and has presented in the RGB color palette (Red, Green, Blue). In this case, the number of matrices presented in the filter must be equal to three.

4.2 Strides

Moving along the matrix of input signals or images is a step that can do with an uneven step of the unit. The step of moving the filter is called the slide.

Stride determines the number of pixels the filter should move in one iteration. Strides [16] should use in cases where, for example, the correlation between adjacent pixels is significant enough. Usually, this observes in the lower layers of the neural network. However, if the value of the stride is too significant, then the inevitable loss of a large amount of information. Also, using the value of the strain, by formula (2), we can calculate the number of initial values after the convolution operation.

$$n_{out} = \lfloor \frac{n_{in} - f}{s} \rfloor + 1 \quad (2)$$

where n_{in} is the number of input pixels, f - the number of pixels at the input of the filter, s - the size of the stride.

4.3 Learning and Activation

In training, the resulting scales in the filter elements can improve. To do this, add offset and non-linearity.

Offset - a statistical value offset from the original values obtained in the convolution process. Otherwise - the mathematical expectation. The offset process is an element-by-element operation of assembling the matrix elements with the amount of shear. The shear rate is immense in the learning process because the initial data are far from the desired result. However, the variance is minimal because the data obtained are not particularly important. At the end of the training, the elimination is minimal since the neural network has identified the main features of the data. We can draw the disappointing conclusion that both small displacement and slight variance are impossible.

Adding non-linearity is done using the activation function, which adds some distortion to the convolution operation. In other words, it determines the activation of neurons in this iteration and the information transmitted to the next layer. In our case, we will use the ReLU activation function, the graph of which shows in Fig. 2.

Effect in Fig. 2 denotes as follows:

- negative input values reset;
- zero does not change;
- positive does not change.

In addition to the ReLU function [18], there are other activation functions, such as sigmoid and hyperbolic tangent. They will not consider in this paper.

4.4 Downsampling

Since the amount of data resulting from the operation of the neural network is vast, algorithms use to reduce the dimension of the source and intermediate data. Such algorithms reduce the size of the data and significantly reduce the number of computing resources required to operate the neural network. One such algorithm is pulling [26].

The idea of pulling is that some fragment contains some strongly expressed properties, so it is possible to refuse a further search of other properties in this area. Thus, considering the matrix of dimension 4×4 , with the help of filters 2×2 and slide dimension 2, we can reduce the amount of output data, at best, to the matrix 2×2 in Fig. 3.

4.5 Loss Function

In the learning process, the neural network should be able to adjust the scales according to the differences between the data obtained from the training. For these purposes, the neural network uses the loss function. Most often, the cross-entropy loss function [8] uses to estimate the accuracy of convolutional neural networks, and it represents by formula (3).

$$H(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i \quad (3)$$

where \hat{y} is the actual response of the neural network, y - required neural network response.

This function determines the actual discrepancy between the two probability distributions. If the value of the function is significant, then the difference between the two distributions is also significant.

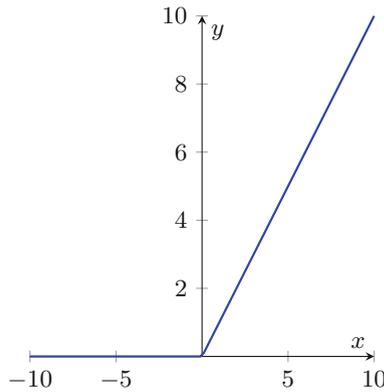


Fig. 2. ReLU activation function graph for $R(z) = \max(0, z)$

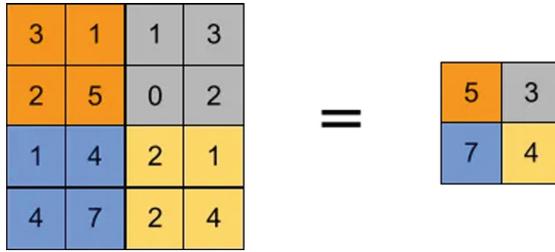


Fig. 3. Illustration of a single pooling layer, for example, max pooling

4.6 Dataset Preparation

After describing all the blocks necessary for building a neural network, the next step is to prepare the datasets and collect the blocks into a model for the neural network. One of the initial stages of building a neural network is the input data description, which describes first below. After that, it carries out a description of the output data and the structure of the neural network.



Fig. 4. Original satellite image for recognition

The original digital satellite image in Fig. 4, which will be used later for recognition, contains the necessary classified objects for recognition in large quantities. First, we need to teach the neural network to recognize the images of the classes we need (trees, water, and fields). We can use similar to the original images and

divide them into different, classified, and masked images. It is better not to use the original image for data preparation, as this will lead to errors in measuring the recognition accuracy.

Additional prepared images, and based on them, images were fragmented to prepare 64×64 pixel datasets. The resulting images divide into three classes, trees, water, and fields, with 20 images for each class.

It is necessary to identify areas that will determine the neural network and belong to the desired class for further training. One way to solve the problem of recognizing similar classes [7] is to use images with butt annotations of the desired class. That allows the neural network to find more unique features and help recognize the class's boundary line, which will better understand its boundaries relative to other classes.

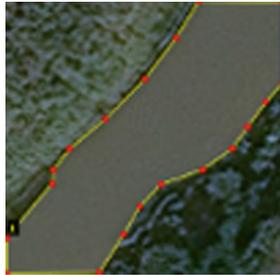


Fig. 5. Annotation of the boundary line between water and its coast

Figure 5 shows an example of an annotation indicating the boundary line between a class of water body and its coast. The line of demarcation is on the border of the main class, which allows us to use for learning the basic features of the class we need and the signs of transition to another class, which allows us to get more accurate results with more input. It will also be more likely to confuse the water body with other similar classes because the neural network will train to find the transition between classes.

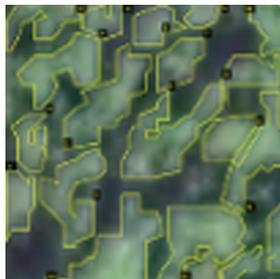


Fig. 6. Annotation of individual trees in the forest

To improve the recognition of more heterogeneous classes, such as the forest, we can use for train the uniform annotation of the forest and its individual, and less unique, objects, such as trees, indicating their boundaries.

Figure 6 shows an example of annotation of individual trees in the forest, the recognition of which we want to teach the neural network. Using images with full tree annotations and images from some unique objects for the class allows us to get more accuracy [13] in recognizing the class itself and, as in the case of water bodies, its boundaries. The main disadvantage of this approach is that it takes more time to annotate the desired class, first and foremost because we need to increase the number of unique features and mark the boundary line.

After annotating the necessary classes and creating masks to follow, the neural network can understand where the class it needs is on the image. Two variants of masks are necessary for different types of research:

- At first, we will not use annotations and the raw version;
- In a second, we will prepare masks on the prepared annotations.

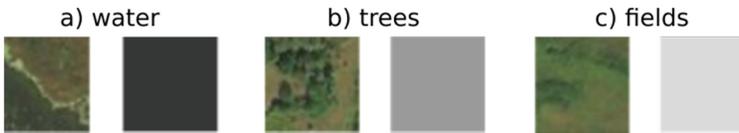


Fig. 7. Raw masks for classes: a) water; b) trees; c) fields

Figure 7 shows the first type with the raw material examples (a class image on the left and a mask on the right) of basic classes such as water, trees, and fields, with masks created for them that neural networks need to understand better where the image is needed.

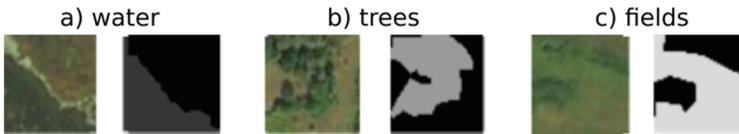


Fig. 8. Annotations based masks for classes: a) water; b) trees; c) fields

Figure 8 shows annotation-based masks, which are processed and have a more detailed allocation of the required area for each class, which allows neural networks to understand more accurately and with fewer mistakes how to recognize each class.

After receiving a dataset ready for training, we can start designing a neural network.

4.7 Neural Network Architecture

First, consider the input layer of the neural network [15]. There should be a prepared dataset at the input: a set of 60 images of different classes (water, trees, and fields), 20 for each class, 48×48 pixels in size, with an RGB color channel. Also, zero-centering normalization applies to the input images, which allows us to subtract the mathematical expectation of the sample from all images.

The following layers will be layers responsible for downsampling, six layers, 3 of which are unique and repeated twice, with the following parameters:

- Convolution layer with a 2×2 filter size in the amount of 50 units. This layer also applies 1-pixel padding to all faces;
- ReLU layer responsible for zeroing negative values;
- Pooling layer with a 2×2 filter size, a stride and vertical and horizontal steps of 3.

The use of previous layers led to a decrease in the input data by a factor of 4, which requires subsequent layers to restore their original sizes. That will require four upsampling layers, 2 of which are unique and also repeated twice:

- Transposed convolutional layer with a filter size of 4×4 , in the amount of 50 units, as well as vertical and horizontal stride steps of 2;
- ReLU layer uses to clip the paddings by 1 pixel.

The last group of output layers is convolutional layers with two 1×1 filters. That is required for linear transformation of the input channels, followed by non-linearity. Further, a layer with the Softmax function uses to obtain the probabilities. This layer transforms the vector into a vector of the same dimension. The elements of the resulting vector are an actual number lying on the segment $[0, 1]$ and interpreted as the probability that a pixel belongs to a particular class. Based on the probabilities, the next step is to label the pixel according to the maximum probability of belonging to a particular class. The pixel classification layer will handle that.

Since the neural network operates with entire data packets at a time, namely, several elements are used at once in the sample, data normalization is required to improve performance and stability. The input of the layer receives a vector of a given dimension $n : x = (x^{(1)}, \dots, x^{(n)})$. We normalize the original vector in each dimension j by formula (4).

$$\hat{x}^{(j)} = \frac{x^{(j)} - Ex^{(j)}}{\sqrt{Dx^{(j)}}} \quad (4)$$

where $x^{(j)}$ is the mean of the vector, $Ex^{(j)}$ is the mathematical expectation of the entire training sample, $Dx^{(j)}$ is the variance of the entire training set.

Considering that normalization performs over the entire training set, it can change the representation of the data within the layer. To accomplish this, we introduce, for each activation $x^{(k)}$, a pair of parameters $\gamma^{(k)}, \beta^{(k)}$, which scale and shift the normalized value by formula (5).

$$\gamma^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{5}$$

Shift and compression parameters select as a result of the learning process. However, in the case of training the model using batches, the batch normalization algorithm [12] is used. It can describe as follows: let the training of a certain model be performed using B packages of a given $m : B = \{x_1, \dots, x_m\}$ dimension. In this case, the normalization will be applied to each input element with the number k separately. Denote the normalization function by formula (6).

$$BN_{\gamma,\beta} : \{x_1, \dots, x_m\} \longrightarrow \{y_1, \dots, y_m\} \tag{6}$$

Then the packet normalization algorithm is presented in Fig. 9. The input algorithm provides the packet values, shear and coagulation parameters γ, β , and the constant ε used for the calculated stability.

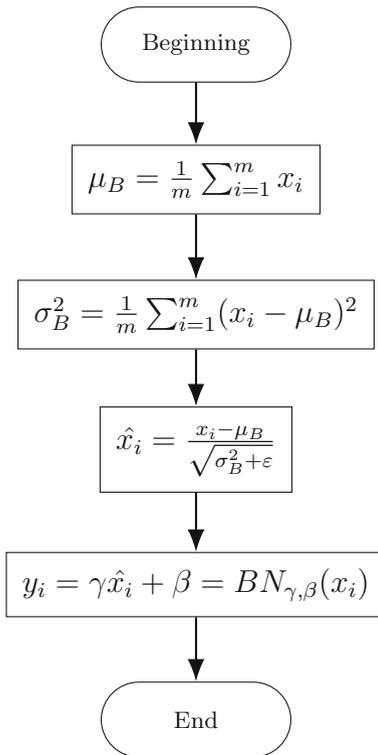


Fig. 9. Batch normalization algorithm

Considering all the parameters of the layers used and the algorithms described above, we can summarize the architecture of the neural network in Table 1.

Table 1. Neural network layer architecture

Number	Name	Parameters
1	Input	$48 \times 48 \times 3$ images with normalization by mathematical expectation
2	Convolution	50 filters with 2×2 convolutions and all edges extended by 1 pixel
3	ReLU	Activation function
4	Max pooling	Pooling 2×2 using the maximum pooling method, a stride (2, 2) and zero paddings
5	Convolution	50 filters with 2×2 convolutions and all edges extended by 1 pixel
6	ReLU	Activation function
7	Max pooling	Pooling 2×2 using the maximum pooling method, a stride (2, 2) and zero paddings
8	Batch normalization	Batch normalization
9	ReLU	Activation function
10	Transposed convolution	50 filters with 4×4 transposed convolutions, a stride (2, 2) and all edges cropping by 1 pixel
11	ReLU	Activation function
12	Transposed convolution	50 filters with 4×4 transposed convolutions, a stride (2, 2) and all edges cropping by 1 pixel
13	ReLU	Activation function
14	ReLU	Activation function
15	Convolution	3 filters with 1×1 convolutions, a stride (1, 1) and zero paddings
16	Softmax	The layer of the probability of belonging to the class
17	Pixel classification layer	Cross-entropy loss

The next step is to determine the learning parameters for the neural network. In verifying the operation of the neural network, the following parameters select empirically. As an optimization function, the chosen method of stochastic gradient descent [11] with the moment describes by formula (7).

$$\theta_{l+1} = \theta_l - \alpha \nabla E(\theta_l) + \gamma(\theta_l - \theta_{l-1}) \quad (7)$$

where l is the iteration number, α is the moment parameter (here it is equal to 0.9), γ is the contribution of the previous gradient descent step in the current iteration.

Adding the coefficient α to the stochastic gradient descent method reduces fluctuations in the learning process. Other parameters are given in Table 2.

Table 2. Neural network learning parameters

Parameter name	Value
Initial learning rate	0.05
Regularization coefficient L_2	0.0001
Max epoch count	150
Validation frequency	30
Training mode	Training according to the plan with the coefficient of “Piecewise-Specified Function”
Shuffle	Every epoch
Gradient threshold method	According to the norm L_2
Gradient threshold	0.05
Package size for training and validation	8

Table 2 shows the parameters for the truncation of the gradient and regularization, the L_2 norm uses as the norm, which indicates by formula (8).

$$|x| = \sum_{i=1}^n \sqrt{|x_i|^2} \tag{8}$$

where x is a vector of dimension $n : x = [x_1 x_2 \dots x_n]^T$.

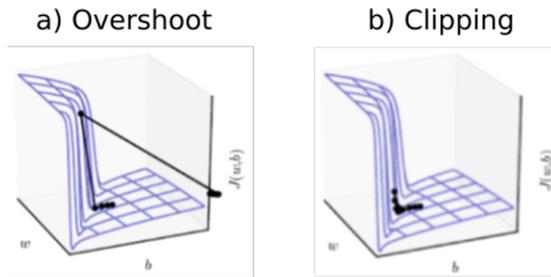


Fig. 10. Gradient descent with: a) Overshoot; b) Clipping

Truncation and normalization of the gradient are required to avoid a large increase in the normal of the gradient during training. An example of such an outlier shows in Fig. 10(a).

As mentioned above, by adding a cut-off, namely - if the gradient rate is greater than the specified value, we assume the resulting rate is equal to the specified upper limit, emissions can avoid. The example shown in Fig. 10(a), with the addition of a clipping, is shown in Fig. 10(b).

5 Experiment, Results and Discussion

After preparing the neural network for training and working with the data we provide, we can start experimenting with different combinations of input data to identify the best combination of time and effort for training and the result. Targets are settings to recognize the provided classes, namely water, trees, and fields. With the help of a neural network, training was carried out on the provided data and checking for the correctness of class recognition. The list of input data as well as the number of required images for all classes show in the Table 3.

Table 3. Ratios of different types of input data for neural network training and further satellite image recognition. The amount of data gives as a ratio of the total.

Training data		Validation data		Number of images	
Raw	Masked	Raw	Masked	Raw	Masked
0.8	0	0.2	0	60	0
0.8	0	0	0.2	48	12
0	0.8	0.2	0	12	48
0	0.8	0	0.2	0	60

Based on the information in Table 3, we can summarize the previous information and come up with the following list:

- 60 images as input data, 20 for each class, following from the information in Subsect. 4.7;
- In total there are 3 classes for recognition and learning: water, trees and fields;
- Each experiment have a different ratio of raw (Fig. 7) and masked (Fig. 8) inputs, which are given in Table 3;
- The ratio of data for training and validation is always the same, namely: 80% for training and 20% for validation;
- Each subsequent experiment contains more input data with masks.

5.1 Experiment 1: Raw Data for Training and Validation

The first experiment involved input images without masks based on annotations. This method of preparing the dataset is the fastest since it requires only prepared class images without selecting the exact area.

As we can see in Fig. 11, with the results of satellite image recognition, only 2 out of 3 classes were recognized: trees and fields. One of the reasons for such poor recognition is that all three classes are pretty challenging to recognize and consistent since most of the pixels of all three classes have many similar colors and primary features. Let us analyze the training progress graph in Fig. 12 to infer the current input for this experiment.

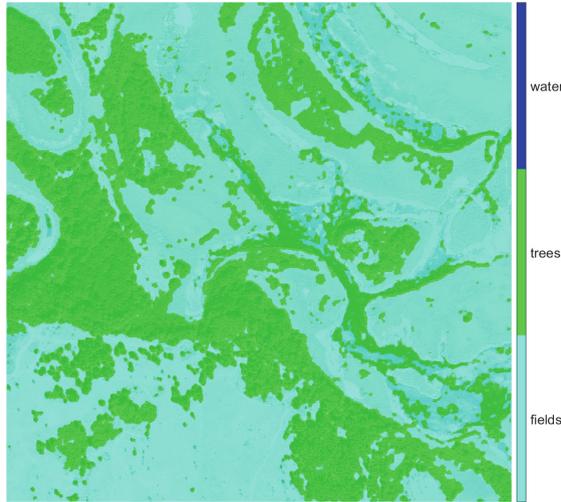


Fig. 11. Recognized satellite image with Training 0.8 (raw) and validation 0.2 (raw)

As can be seen in the graph in Fig. 12, *Mini-batch* and *Validation* (85.31%) accuracies are quite high, which is an overestimation when looking at the results in Fig. 11. That is due to using the same set of input data for neural network training and validation during training.

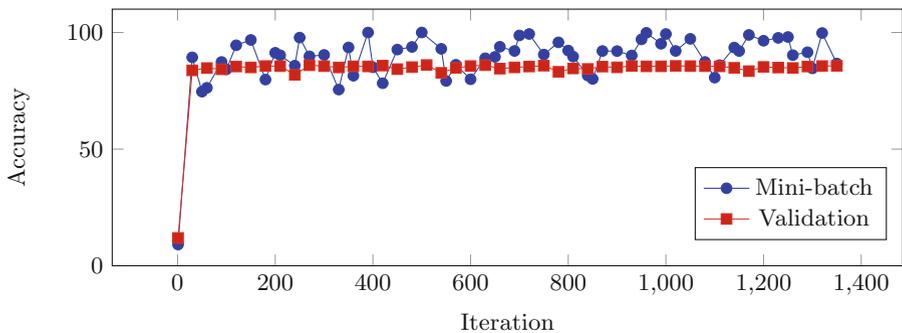


Fig. 12. Graph with training progress of Experiment 1.

Summing up the results of Experiment 1, we can say that using the current set of input data, namely raw training and validation data, is not suitable for working on Fig. 4 recognition.

5.2 Experiment 2: Raw Data for Training and Masked for Validation

The second experiment involved raw input training and masked validation data. This method requires the preparation of pre-annotation and subsequent masking

for 20% of the input data that will use for validation, and the training data remains raw.

Satellite image recognition results from Fig. 4 using masked validation and raw training input data in Fig. 13.

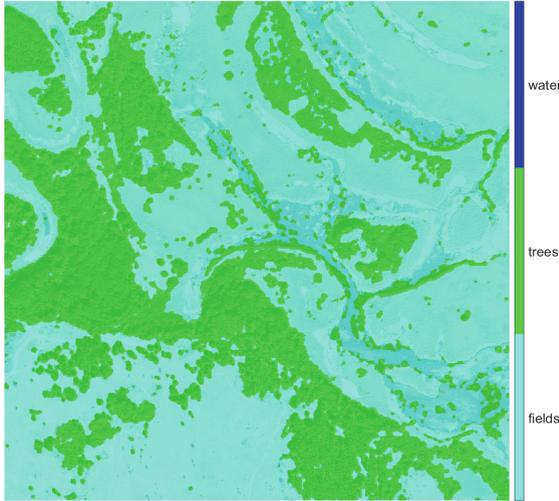


Fig. 13. Recognized satellite image with Training 0.8 (raw) and validation 0.2 (masked)

As we can see in Fig. 13, with the results of satellite image recognition, still only 2 out of 3 classes were recognized: trees and fields. Let us analyze the training progress of the experiment in Fig. 13 to summarize.

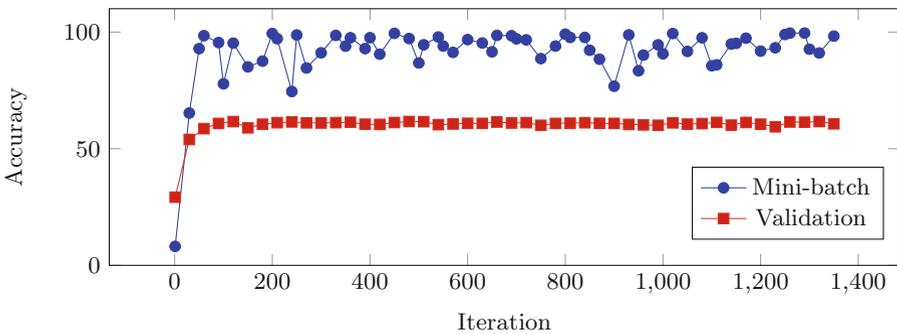


Fig. 14. Graph with training progress of Experiment 2.

The results of the graph in Fig. 14 show that *Mini – batch* accuracy is quite close to the results of Experiment 1 in the graph from Fig. 12. However, at the

same time, the results of *Validation* accuracy dropped quite significantly, from 85.31% in Experiment 1 to 61.55%, which makes us convinced of the low-quality recognition.

Summing up the results of Experiment 2, we can say that using the current set of input data, namely raw training, and masked validation data, is still not suitable for working on Fig. 4 recognition.

5.3 Experiment 3: Masked Data for Training and Raw for Validation

The third experiment involved masked input training and raw validation data. This method requires preliminary preparation of the annotation and subsequent masking for 80% of the training data, and the validation ones remain raw.

Satellite image recognition results from Fig. 4 using raw validation and masked training input data in Fig. 15.

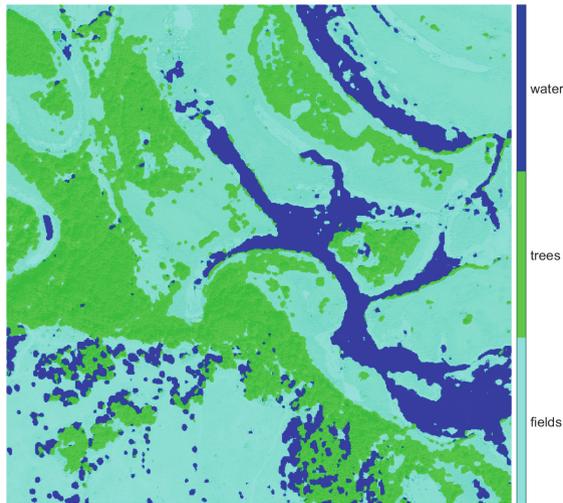


Fig. 15. Recognized satellite image with Training 0.8 (masked) and validation 0.2 (raw)

As we can see in Fig. 15, with the results of satellite image recognition, all the classes recognize success. Let us analyze the learning progress in the graph in Fig. 16 to conclude the current results.

The results of the graph in Fig. 16 shows that *Mini - batch* and *Validation* (84.10%) accuracies are high and reasonably similar to the results from Experiment 1 in the graph from Fig. 13. However, the difference in the result itself in Fig. 15 is significant since, unlike Experiment 1, all classes recognize success.

Summing up Experiment 3, we can say that using marked training data allowed us to achieve better results than using raw training data in Experiments 1 and 2, regardless of whether the validation data was raw or masked.

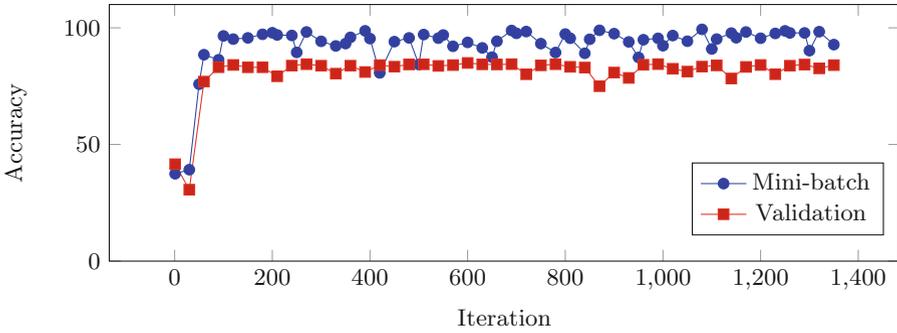


Fig. 16. Graph with training progress of Experiment 3.

5.4 Experiment 4: Masked Data for Training and Validation

The fourth experiment involved input only masked data for training and validation. This method requires preliminary preparation of the annotation and subsequent masking for training and validation data, making the data preparation process more difficult than Experiment 3.

Satellite image recognition results from Fig. 4 using masked training and validation input data in Fig. 17.

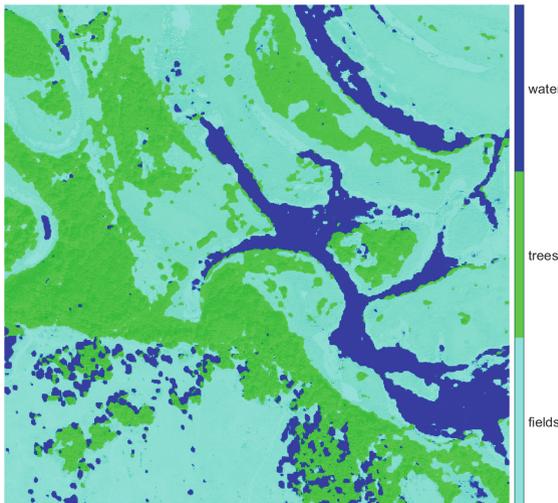


Fig. 17. Recognized satellite image with Training 0.8 (masked) and validation 0.2 (masked)

As in the case of Experiment 3, all classes recognize success, and it is also worth noting that the results at first glance look pretty similar. Let us analyze the training progress in the graph in Fig. 18 for comparison with Experiment 3.

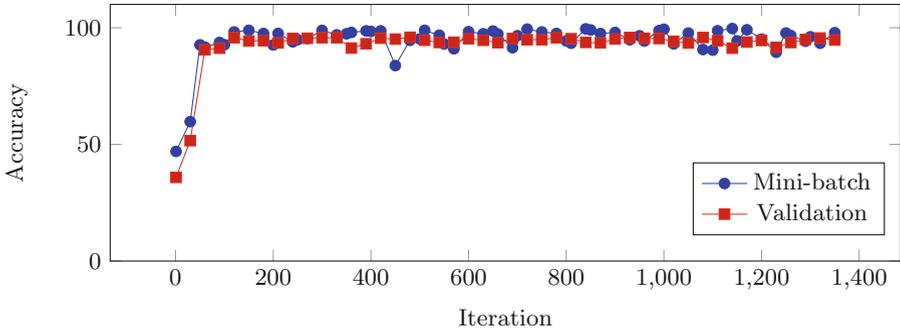


Fig. 18. Graph with training progress of Experiment 4.

The results of the graph in Fig. 18 shows that *Mini – batch* and *Validation* (95.06%) accuracies are almost identical. Difference with Experiment 3 in the graph from Fig. 16 is in *Validation* accuracy, which is up 10.96%.

Summing up from Experiment 4, improving the quality of the *Validation* data could not significantly increase the advantage in recognition quality compared to Experiment 3. Also, compared with the second experiment, the recognition accuracy increased by 54.44% (comparison with Experiment 2 is due to the same validation data there.).

6 Conclusions

The research on evaluating the impact of input data processing on the quality of neural network training for satellite image recognition has confirmed the assumption about the importance of careful preparation of input data.

An input dataset was prepared for training to recognize three classes: water, trees, and fields. The dataset containing 120 prepared images and 60 were pre-annotated and created masks, with an individual approach to each class. The other 60 contained a uniform mask without a precise selection of class elements. The ratio of training data to validation data was 80% to 20%.

Figure 19 shows the increase in the quality of class recognition. In comparing Experiments 1 and 4 (Experiment 2 had a similar result to 1 and 3 to 4), the increase was 54.44%.

Summing up the results of all four experiments, we can say that using a convolutional neural network and prepared training data has satisfactory results in classifying and recognizing a multi-spectral satellite image of ultra-high resolution. The classification accuracy increase requires considering a promising sample of a larger dimension and highlighting the class boundaries in more detail

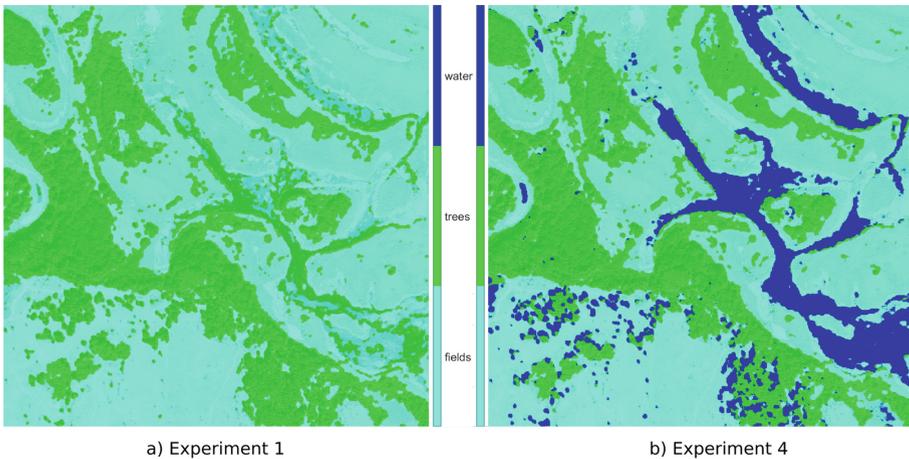


Fig. 19. Results of the first and fourth experiments

when preparing masks. Another modification option may be to consider using other channels for the training set. Also, since the selection of boundaries for the dataset at this stage of development was manual, the human factor should be considered.

The further perspectives of the authors' research are the implementation of edge detection on the dataset using automatic methods, followed by human correction, and automation in the selection of input values for the neural network architecture, which will significantly increase the number of experiments and achieve better results.

References

1. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6. IEEE (2017). <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
2. Ali, N., Zafar, B., Iqbal, M.K., et al.: Modeling global geometric spatial information for rotation invariant classification of satellite images. *PLoS ONE* **14**(7), e0219833 (2019)
3. Arcidiacono, C., Porto, S.: Classification of crop-shelter coverage by RGB aerial images: a compendium of experiences and findings. *J. Agric. Eng.* **41**(3), 1–11 (2010)
4. Hirayama, H., Sharma, R., Tomita, M., Hara, K.: Evaluating multiple classifier system for the reduction of salt-and-pepper noise in the classification of very-high-resolution satellite images. *Int. J. Remote Sens.* **40**(7), 2542–2557 (2019). <https://doi.org/10.1080/01431161.2018.1528400>

5. Hnatushenko, V., Zhernovyi, V.: Method of improving instance segmentation for very high resolution remote sensing imagery using deep learning. In: Babichev, S., Peleshko, D., Vynokurova, O. (eds.) DSMP 2020. CCIS, vol. 1158, pp. 323–333. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61656-4_21
6. Hordiiuk, D.M., Hnatushenko, V.V.: Neural network and local Laplace filter methods applied to very high resolution remote sensing imagery in urban damage detection. In: Proceedings of 2017 IEEE International Young Scientists Forum on Applied Physics and Engineering (2017). <https://doi.org/10.1109/ysf.2017.8126648>
7. Hosseini, H., Xiao, B., Jaiswal, M., Poovendran, R.: On the limitation of convolutional neural networks in recognizing negative images. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 352–358. IEEE (2017). <https://doi.org/10.1109/ICMLA.2017.0-136>
8. Hu, K., Zhang, Z., Niu, X., et al.: Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function. *Neurocomputing* **309**, 179–191 (2018). <https://doi.org/10.1016/j.neucom.2018.05.011>
9. Ji, Z., Telgarsky, M.: Directional convergence and alignment in deep learning. *Adv. Neural. Inf. Process. Syst.* **33**, 17176–17186 (2020)
10. Julio, O., Soares, L., Costa, E., Bampi, S.: Energy-efficient gaussian filter for image processing using approximate adder circuits. In: 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 450–453. IEEE (2015). <https://doi.org/10.1109/ICECS.2015.7440345>
11. Ketkar, N.: Stochastic gradient descent. In: Deep Learning with Python, pp. 111–130. Apress, Berkeley (2017). https://doi.org/10.1007/978-1-4842-2766-4_8
12. Kohler, J., Daneshmand, H., Lucchi, A., et al.: Towards a theoretical understanding of batch normalization. *Stat* **1050**, 27 (2018)
13. Mozgovoy, D., Hnatushenko, V., Vasyliiev, V.: Accuracy evaluation of automated object recognition using multispectral aerial images and neural network. In: Proceedings of the SPIE 10806, Tenth International Conference on Digital Image Processing (2018). <https://doi.org/10.1117/12.2502905>
14. Mozgovoy, D.K., Hnatushenko, V.V., Vasyliiev, V.V.: Automated recognition of vegetation and water bodies on the territory of megacities in satellite images of visible and IR bands. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **IV-3**, 167–172 (2018). <https://doi.org/10.5194/isprs-annals-IV-3-167-2018>
15. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: a deep neural network architecture for real-time semantic segmentation. arXiv preprint [arXiv:1606.02147](https://arxiv.org/abs/1606.02147) (2016). <https://doi.org/10.48550/arXiv.1606.02147>
16. Riad, R., Teboul, O., Grangier, D., Zeghidour, N.: Learning strides in convolutional neural networks. arXiv preprint [arXiv:2202.01653](https://arxiv.org/abs/2202.01653) (2022). <https://doi.org/10.48550/arXiv.2202.01653>
17. Sader, S., Bertrand, M., Wilson, E.H.: Satellite change detection of forest harvest patterns on an industrial forest landscape. *Forest Sci.* **49**(3), 341–353 (2003)
18. Schmidt-Hieber, J.: Nonparametric regression using deep neural networks with ReLU activation function. *Ann. Stat.* **48**(4), 1875–1897 (2020). <https://doi.org/10.1214/19-AOS1875>
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. [arxiv:1409.1556](https://arxiv.org/abs/1409.1556), September 2014 (2020). <https://doi.org/10.48550/arXiv.1409.1556>

20. Suganuma, M., Shirakawa, S., Nagao, T.: A genetic programming approach to designing convolutional neural network architectures. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 497–504 (2017). <https://doi.org/10.1145/3071178.3071229>
21. Tokarev, K.: Intelligent system for agricultural crops defective areas monitoring and visualization based on spectral analysis of satellite high-resolution images. In: IOP Conference Series: Earth and Environmental Science, vol. 786, p. 012039. IOP Publishing (2021)
22. Wu, F., Wang, Z., Zhang, Z., et al.: Weakly semi-supervised deep learning for multi-label image annotation. *IEEE Trans. Big Data* **1**(3), 109–122 (2015)
23. Xu, J.L., Gowen, A.: Spatial-spectral analysis method using texture features combined with PCA for information extraction in hyperspectral images. *J. Chemom.* **34**(2), e3132 (2020). <https://doi.org/10.1002/cem.3132>
24. Yu, Z., Li, T., Luo, G., Fujita, H., Yu, N., Pan, Y.: Convolutional networks with cross-layer neurons for image recognition. *Inf. Sci.* **433**, 241–254 (2018). <https://doi.org/10.1016/j.ins.2017.12.045>
25. Zhang, X., Wang, H., Hong, M., et al.: Robust image corner detection based on scale evolution difference of planar curves. *Pattern Recogn. Lett.* **30**(4), 449–455 (2009). <https://doi.org/10.1016/j.patrec.2008.11.002>
26. Zhang, Y., Zhao, D., Zhang, J., Xiong, R., Gao, W.: Interpolation-dependent image downsampling. *IEEE Trans. Image Process.* **20**(11), 3291–3296 (2011). <https://doi.org/10.1109/TIP.2011.2158226>



Application of Wavelet Transform for Machine Learning Classification of Time Series

Lyudmyla Kirichenko^{1,2} , Oksana Pichugina^{3,4} , Tamara Radivilova¹ ,
and Kyrlo Pavlenko¹ 

¹ Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
{lyudmyla.kirichenko,tamara.radivilova,kyrylo.pavlenko}@nure.ua

² Wroclaw University of Science and Technology, Wroclaw, Poland

³ National Aerospace University “Kharkiv Aviation Institute”, Kharkiv, Ukraine
o.pichugina@khai.edu

⁴ University of Toronto, Toronto, Canada

Abstract. The work is devoted to the time series classification based on the application of continuous wavelet transform and visualization of the resulting wavelet spectrum. Wavelet spectrum images are input to a neural network that classifies them. Wavelet transform allows one to analyze the time variation of frequency components of time series. The paper considers the classification of time realizations subject to normal additive noise with different variances. The wavelet spectrum visualization for various wavelet functions is presented. A residual neural network was used for the classification of the spectra images. The computational experiment results give evidence that the classification based on the recognition of wavelet spectra images allows qualitative distinguishing signals with an additive noise component having different signal-to-noise levels. Thus, we recommend applying the proposed method for classifying noisy time series of different types, such as medical and biological signals, financial time series, information traffic and others.

Keywords: Time series classification · Continuous wavelet transform · Wavelet-spectrum image · Deep learning

1 Introduction

A time series (TS) is a set of data points ordered in time [4], i.e. it is a numerical sequence. Most commonly, a TS is a sequence of values taken in equidistant timing. Time Series Analysis (TSA) is a branch of mathematics and machine learning that studies methods for exploring behavior of time series by extracting and analyzing their internal characteristics.

The dynamics of real complex systems of various types, such as mechanical, social, and informational, are commonly represented by time series. The study of time series nature leads to a deep understanding of the essence of underlying

processes. As a result, it allows for solving many problems, such as classification ones. Over the past decade, Machine Learning (ML) methods have become a promising alternative to traditional time series classification methods. The ML approach is highly attractive because it allows for classifying datasets without introducing analytic rules. ML allows deriving these rules directly from the data. On the other hand, ML requires a relatively large number of observations to train classifiers, which is far from always possible. Therefore, simulated time realizations play an important role in ML, making it possible to test the performance of classification algorithms for the further transition to the analysis of real-world time series.

2 Literature Survey

TSA methods are divided into two broad classes – frequency-domain methods (FD methods) and time-domain methods (TD methods). In turn FD methods are divided into Spectral Analysis and Wavelet Analysis [22, 28]; while, among TD methods, Auto-correlation Analysis and Cross-Correlation analysis can be singled out [4, 12].

Numerous problems are solved in TSA including forecasting and classification. Time series classification (TSC) deals with assigning time series to specific categories (classes) thus predicting actual categories of TSs [21]. Among practical applications of TSC are: electronic health records, human activity recognition, acoustic scene classification, cybersecurity, speech recognition, financial analysis, manufacturing, power systems [13–16, 23–25, 27] etc.

The importance of effectively solving this class of classification problems has grown recently due to the increase of time data availability [6]. At the same time, TSC is considered as a highly challenging problems in data mining [7, 29]. Main complicacy and distinction TSC from standard classification dealing with tabular classifiers is that datapoints are ordered, disregarding if the ordering in time or not. This means that a TS can possess some specific features depending on the ordering. The peculiarities have to be taken into account in classification. This challenging fact inspired many researches to make contribution in developing TSC methods. To date, hundreds of methods have been presented in the literature [3, 21, 27]. The following typology of TSC algorithms can be offered [21, 27]: a) Distance-based, e.g. KNN with dynamic time warping; b) Interval-based, e.g. TimeSeriesForest; c) dictionary-based, such as BOSS, cBOSS; d) Frequency-based, for in-stance RISE; e) Shapelet-based, e.g. Shapelet Transform Classifier.

One of the powerful tools for analyzing time series, especially non-stationary ones, is applying discrete and continuous wavelet transforms [28]. The utilization of wavelet coefficients for the classification of time series is considered in many recent works [2, 19, 26, 30].

Deep learning is also popular and highly promising tool to perform TSC [6, 11, 13, 20, 24, 25]. For instance, experimentally it was justified that long short-term memory neural networks (LSTMs) and convolutional neural networks (CNNs) are capable to mine dynamical characteristics of time series [23]. Still

many open questions in applying neural networks in TSA such as a choice of an efficient neural network (NN) architecture; how effectively tune NN-parameters (numerical and related to NN architecture), a NN training requires large amount of samples that is not always available, training of any NN normally requires more time than other classification algorithms.

One of the undoubted achievements of deep learning is higher accuracy in image classification, especially if convolutional and residual neural networks are used [5, 10]. Many of them are presented in the form of ready-made models and are available for utilization. Thus, a quite promising approach to the classification of time series is their transformation into a certain structure well suited for visualization, followed by classification by deep neural networks. Such an approach based on the analysis of recurrence plots was considered in [17, 18]. A similar approach is to visualize the result of the wavelet transform as a wavelet spectrum image and further classify the obtained images [8, 9].

The presented work is dedicated to classifying simulated time series with an additive noise component, where different noise levels characterize classes. The work aims to study the classification of noisy time series based on an analysis of their wavelet spectra. The spectra are planned to be used as input data for a deep neural network classifier.

3 Materials and Methods

3.1 Continuous Wavelet Transform in Time Series

Normally, the real-world time series to be analyzed are non-stationary, noisy and heterogeneous. The wavelet transform was developed in the mid-80s as an alternative to the Fourier transform for studying strongly degenerated time series. The harmonic basis functions of the Fourier transform are extremely localized in the frequency domain and not localized in the time domain. In terms of localization, wavelets can be considered as functions occupying an intermediate position between harmonic and impulse functions.

Since wavelet transform possesses a movable time-frequency window, it equally qualitatively detects low- and high-frequency signal characteristics on different time scales. Wavelet analysis is especially effective for the study of time series including segments with different behavior patterns. Wavelet transform of a one-dimensional signal is its representation in the form of a generalized series or integral over a system of basis functions:

$$\psi_{ab}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right),$$

obtained from the mother wavelet $\psi(t)$. The wavelet possesses certain properties due to the operations of translating b and dilating a in time. The factor $\frac{1}{\sqrt{a}}$ ensures that the norm of these functions is independent from the dilation parameter a .

A function $\psi : \mathbb{R} \rightarrow \mathbb{L}^2$ is called a wavelet if it satisfies the following constraints $\psi \in \mathbb{L}^2, \|\psi\| = 1$,

$$C_\psi = 2\pi \int_{\mathbb{R} \setminus \{0\}} \frac{|S_\psi(w)|^2}{|w|} dw < \infty,$$

where $S_\psi(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(t) \cdot e^{-iwt} dt$.

The continuous direct and inverse wavelet transform $X(t)$ of the signal has the form of

$$X(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(a, b) \psi_{ab}(t) \frac{da \cdot db}{a^2},$$

where $\langle \cdot, \cdot \rangle$ is the dot product operation, $\psi_{ab}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$.

Various compactly supported functions can play a role in generating wavelet functions. They have to be bounded in time and location on the time axis and have a spectral image, to a certain extent, localized on the frequency axis.

The choice of the wavelet to be analyzed is largely determined by the type of information that needs to be extracted from the signal. Taking into account typical features of various wavelets in time and frequency space, it is possible to identify certain properties and characteristics of the analyzed signals that are invisible on the signal graphs, especially if strong noise is present.

To construct wavelets, derivatives of Gaussian functions are often used. The functions have the best localization in time and frequency domains. The most famous and popular in this family is the wavelet WAVE, which is calculated from the first derivative of the Gaussian function. Another is the wavelet MHAT (Mexican hat), which is calculated from the second derivative. The wavelet MHAT is symmetric, its spectrum is well localized in frequency, and the wavelet zero- and first moments are equal to zero. MHAT is usually used to analyze complex signals. Figure 1 depicts the wavelets WAVE and MHAT in the time domain.

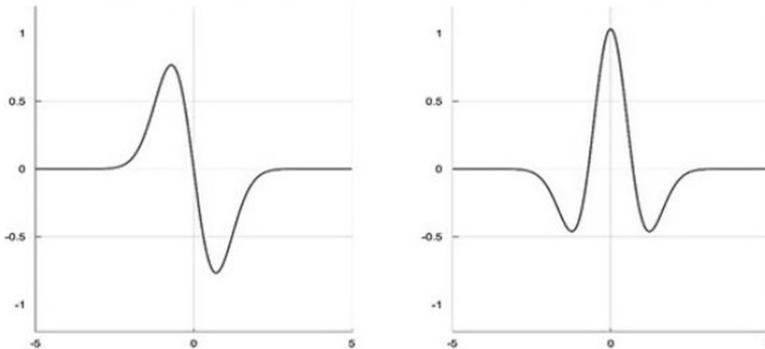


Fig. 1. Wavelets WAVE and MHAT

With an increase in the number of the derivative of the function, the time domain of the wavelet definition slightly increases with a sufficiently significant increase in the dominant frequency of the wavelet and the degree of its localization in the frequency domain. n -th order wavelets enable a finer analysis of high-frequency signal structures by suppressing low-frequency components.

The result of a wavelet transform of a time series is a two-dimensional array $W(a, b)$ of values of coefficients if the wavelet spectrum. The distribution of wavelet spectrum values on the (a, b) -plane provides information about the change in time of the relative contribution to the signal of wavelet components of different scales.

$W(a, b)$ is a surface in 3D space. Spectrum visualization methods can be chosen differently. The most common method is the projection onto the plane (a, b) with isolines. This way makes it possible to trace changes in the coefficients in different time scales and reveal patterns in these surfaces' local extrema.

On the vertical sections of the translation parameter b , the wavelet spectrum reflects the component composition of the signal at each time moment. Since the values of $W(a, b)$ are calculated as dot product of a signal and a wavelet. Evidently, values of the coefficients at each current time point along the scale cuts are the greater, the stronger correlation between the wavelet of a given scale and the behavior of the signal in the vicinity of this point. Scale cuts for the parameter a demonstrate changes in a signal of the component of a given scale a with time.

When analyzing signals with the help of wavelets, harmonic signals normally correspond to bands of wavelet peaks and troughs at dominant wavelet frequencies coinciding with the frequency of signal harmonics. The more clear signal features, the stronger they stand out on the spectrograms. Noise components are usually small-scale and located in the lower part of the wavelet spectrum. Figure 2 shows the wavelet spectra of a noisy sinusoid obtained by wavelets MHAT, Gauss.8 and Morlet.

3.2 Input Data

One of the important tasks in signal processing and classification of time series are the problems related to noisy time realizations. Extracting and removing noise components of signals is one of the strengths of wavelet processing. Therefore, sinusoidal time realizations with and without additive normal noise were chosen for our numerical experiments.

We apply two different approaches to splitting time realizations into classes. In the first case, the classes are noisy time realizations of sinusoids with frequencies lying in different ranges. In the second, all time realizations are divided into two classes: noisy and unnoisy.

Let us consider both cases in more detail.

Each class is a noisy sinusoid with different frequencies in the first case. A frequency range is specified for each class, while the ranges are not pairwise disjoint. The frequency was chosen randomly from the corresponding range when

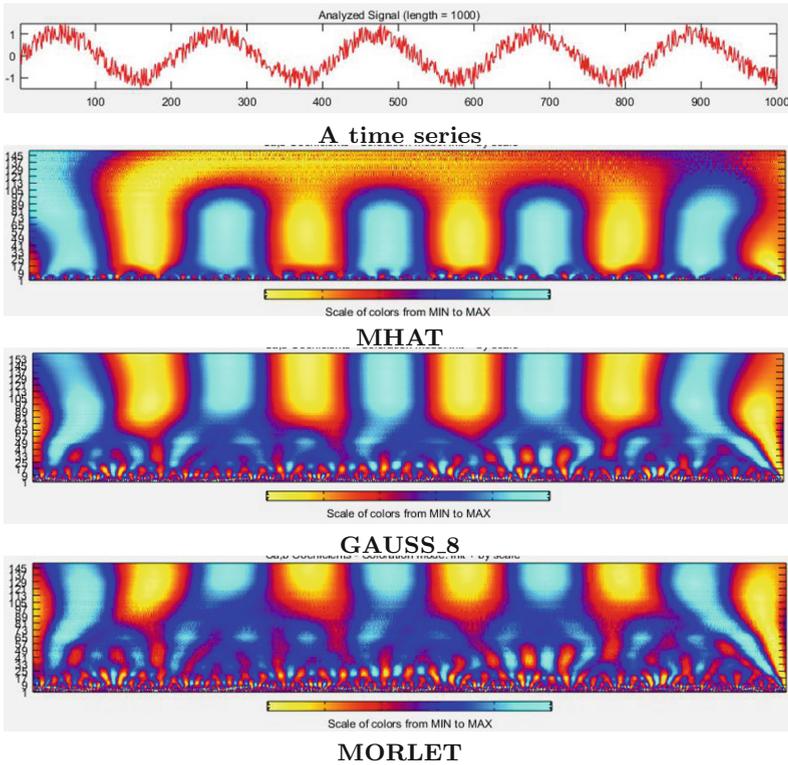


Fig. 2. Wavelet spectra of a noisy sinusoid obtained using different wavelets

generating the time realization of S . For noisy time realizations of S_{noise} , normal distribution noise $N(0, \sigma)$ with zero mathematical expectation and a given standard deviation σ were used.

$$S_{noise} = S + Noise, \tag{1}$$

where $Noise$ is a realization of independent values of a normal random variable.

Signal-to-noise ratio is evaluated as the ratio of standard deviations σ_S and σ_{noise} of signal and noise, respectively:

$$R = \frac{\sigma_S}{\sigma_{noise}} \tag{2}$$

was used as a parameter for studying the quality of classification.

The smaller this ratio, the more noisy time realizations.

In the second case, the first class represents non-noisy sinusoids, and the second one includes signals where part of its time realizations are noisy. Additive noise (1) was added at a randomly selected time realization interval, no less than 0.25 of the entire value. The frequency of the sinusoid was chosen randomly from a given range.

3.3 Classification Metrics

The paper considers binary classification problems for which such classification quality metrics as Accuracy, Precision, Recall, F_1 -score and ROC AUC are widely used.

These metrics are calculated based on utilizing the number of correctly and falsely detected cases of different classes presented in the form of error matrix: true positives (**TP**), true negatives (**TN**); false positives (**FP**) and false negatives (**FN**).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

shows the proportion of correctly classified objects.

$$Precision = \frac{TP}{TP + FP}$$

indicates the proportion of features that are defined as positive and are actually positive,

$$Recall = \frac{TP}{TP + FN}$$

shows what proportion of objects in a positive class out of all objects in a positive class that were correctly detected.

F_1 -score (F_1 -measure) is a metric

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

equal to the harmonic mean of precision and recall.

Also, the assessment of the classification quality as a whole is the Area Under Curve (**AUC**) – Receiver Operating Characteristic (**ROC**) curve. The ROC curve is the one given in the range from (0,0) to (1,1) in True Positive Rate (**TPR**) and False Positive Rate (**FPR**) coordinates,

$$TPR = \frac{TP}{TP + FN},$$

$$FPR = \frac{FP}{FP + TN}$$

and take values from 0.5 to 1.

4 Experiment

The classification algorithm is outlined in Figs. 3 and 4. It consists of two stages. At the first stage, each time realization from sample or test data sets was transformed into a wavelet spectrum using a continuous wavelet transform (Fig. 3). At the second stage, the obtained two-dimensional images were fed to the input of our neural network, by which they were classified (Fig. 4).

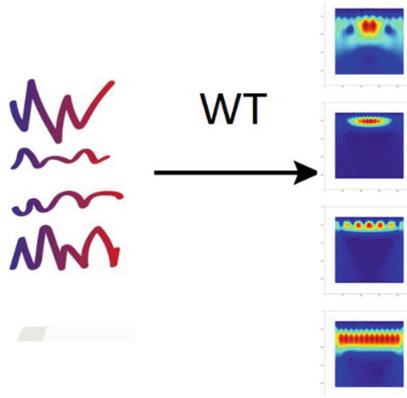


Fig. 3. Formation of a data set of wavelet spectra images

Note that the division into two stages is conditional and, in the case under consideration, it simply emphasizes the importance of the preliminary choice of a wavelet function and frequencies of processing of the signal. In addition, it is possible to form several datasets of wavelet spectra obtained by implementing different wavelet functions for the same time data. This allows analyzing the classification quality for different wavelets and choosing the most appropriate one.

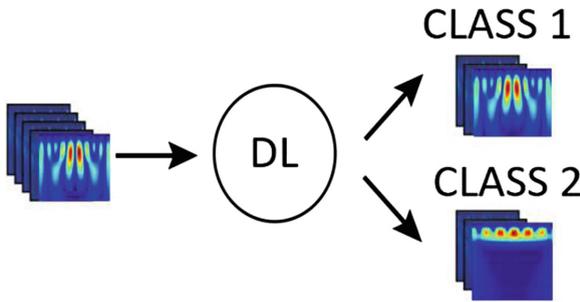


Fig. 4. Image classification based on deep neural network implementation

The second stage is the actual classification of wavelet spectra images. The success of applying deep neural networks in image recognition is widely known and the off-the-shelf neural network models.

Residual networks are used in image recognition tasks. They allow transferring the information between the layers without fading due to the usage of short links. Increasing the number of layers is necessary to find ideal weights and thresholds (improving network performance) and calculate loss functions and back propagations (finding and multiplying derivatives of the whole chain). The vanishing/exploding gradient problem arises. To solve it, residual networks

use identity shortcut connections, which are contained in residual blocks, skip training one or more layers, and connect directly to the output. The residual block teaches the output of the previous residual block, that is, the residual function. In this way, each block trains the output of the previous block.

In this paper, classification is performed using a convolutional neural network with a depth of 50 layers, called ResNet-50 [1].

5 Results and Discussion

5.1 Time Realizations and Wavelet Spectra

On Fig. 5, noisy realizations of two classes with different R -values are shown. On the left, the sinusoids of the class with a lower frequency range are illustrated, and on the right, with higher ones. The signal-to-noise ratio decreases from top to bottom

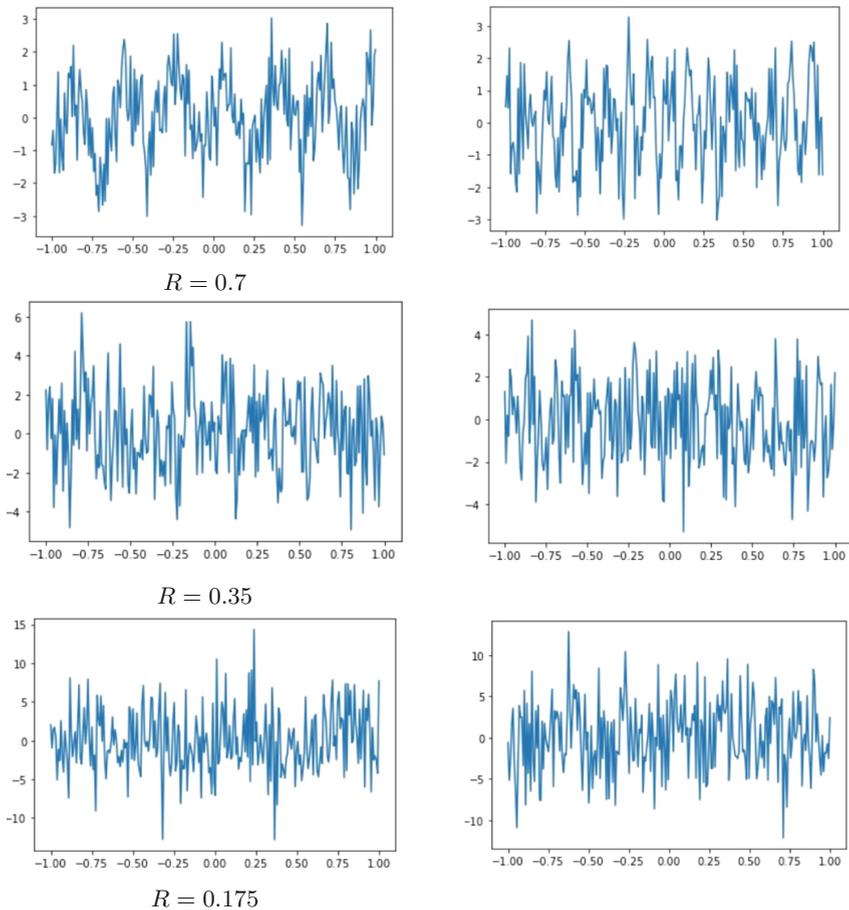


Fig. 5. Noisy sinusoids from two classes with different signal-to-noise ratios

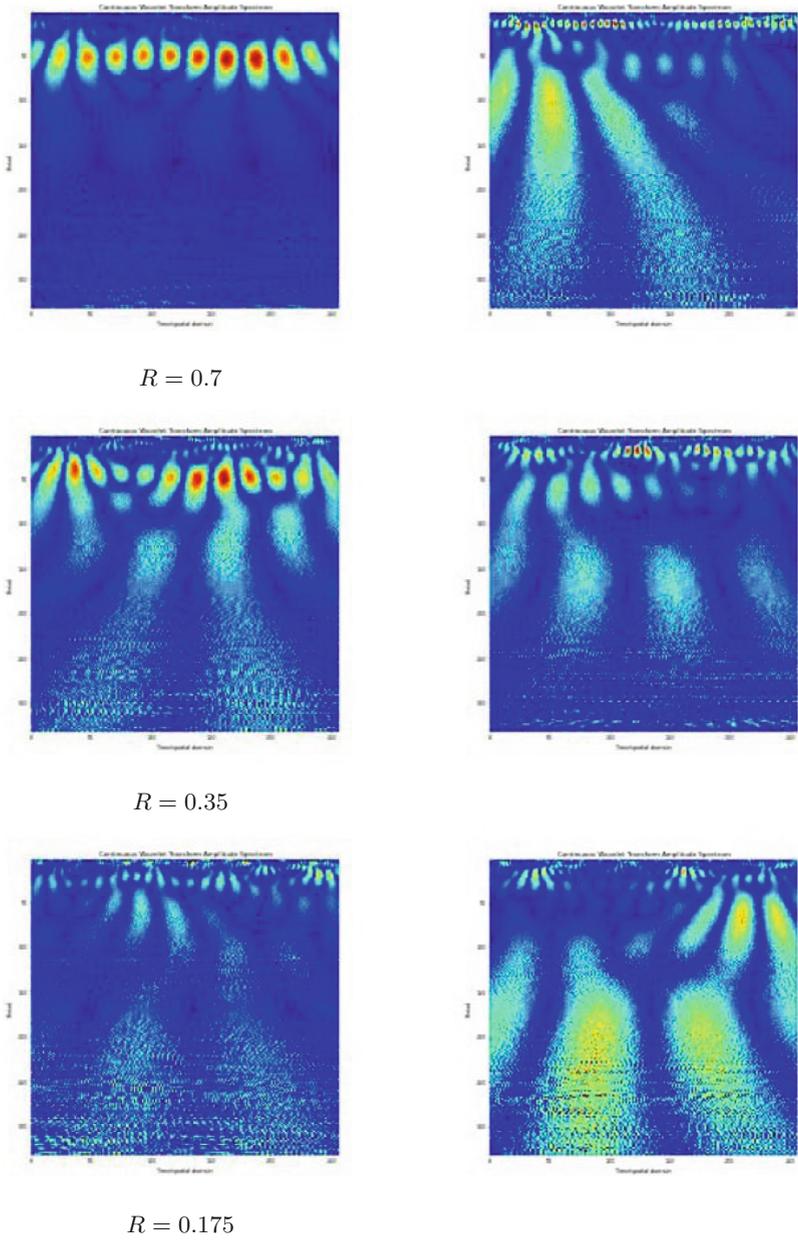


Fig. 6. Wavelet spectra of noisy signals with different signal-to-noise ratios

Figure 6 shows the corresponding wavelet spectra for the above noisy signals. It should be noted that visually the spectra differ more than the time realizations. Figure 7 depicts some typical noisy time realizations

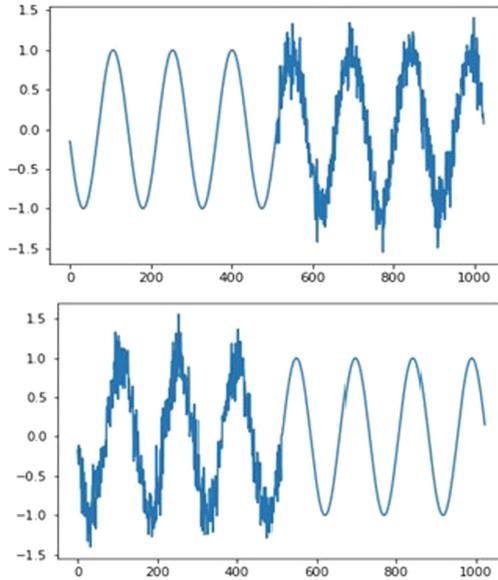


Fig. 7. Partially noisy time realizations

The signal-to-noise ratio (2) was calculated as a ratio of a standard deviations of the signal σ_S and noise σ_{noise} . It is used as a class discrimination parameter. The smaller the value, the smaller the difference between clean and noisy signals. For a fixed standard deviation of the signal σ_S , the difference between signal classes and classification accuracy decreases as the parameter R increases.

Figure 8 illustrates one of the partially noisy realizations (a) and its wavelet spectra (b) obtained by utilizing different wavelet functions: Mexican Hat (left), 1-st order Gaussian wavelet (middle), and Morlet wavelet (right). Evidently, the choice of the wavelet function significantly affects quality of recognition of the image.

In both classification problems, the training samples consist of 350 items, and the test samples contain 150 items, with the same number of items in each class. Consider the classification problem for data of the first type where the classes were noisy time realizations with frequencies from different ranges. Preliminary studies were carried out on the choice of wavelet functions and an estimate classification based on implementing different wavelet spectra. Table 1 shows the preliminary results of evaluating the classification accuracy, wherefrom it is seen that it is advisable to use the Mexican hat wavelet in this case.

Table 1. Classification accuracy for different wavelets

Metric	1st order Gauss	Mexican Hat	8th order Gauss	Morlet
Accuracy	0.95	0.96	0.95	0.92

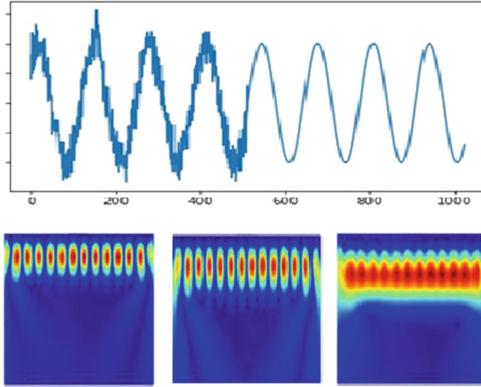


Fig. 8. Noisy time realization and its wavelet spectra

One of the goals of implementing classification techniques for time series analysis is to explore how much time realizations can be noised in order to achieve acceptable accuracy of classification. Thus, it is necessary to investigate the dependence of the classification accuracy on the signal-to-noise ratio (2). Figure 9 shows ROC curves for $R = 0.35, 0.175$ and 0.115 .

Table 2 presents quality metrics depending on values of the coefficient R

Table 2. Classification metrics for TS-samples with frequencies from different ranges

R	Accuracy	Precision	Recall	f1 score	ROC AUC
0.7	1.00	1.000	1.000	1.0000	1.0000
0.35	0.96	0.92	0.99	0.96	0.99
0.175	0.7	0.71	0.68	0.69	0.779
0.115	0.58	0.572	0.58	0.576	0.637

Thus, the classification accuracy remains high at sufficiently small values of the signal-to-noise ratio R (see Fig. 8). This allows us to recommend the use of the proposed method for the classification of noisy time series of different types.

5.2 Classification of Partially Noisy Sinusoids

Consider classification of time series of the second type, where the classes were pure realizations or partially noisy ones. The Mexican hat wavelet was also

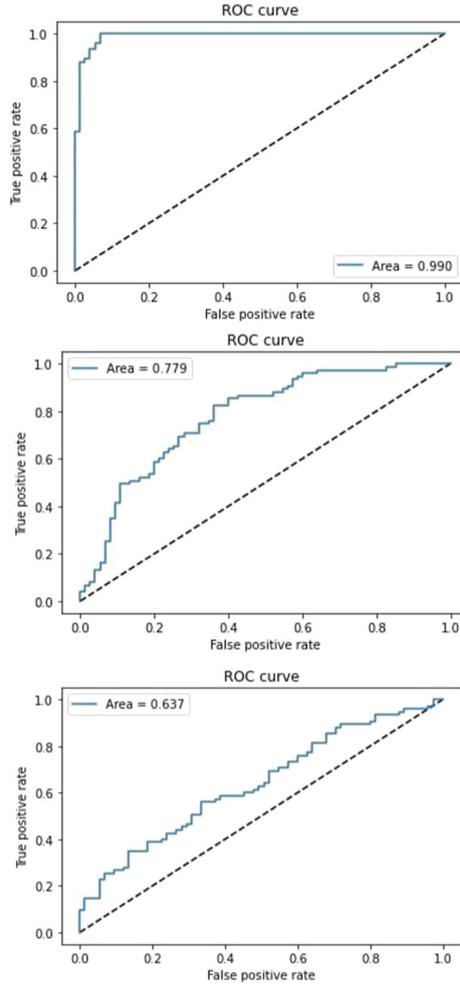


Fig. 9. ROC curves for $R = 0.35, 0.175, 0.115$ (from top to bottom)

chosen as a wavelet function. Table 3 shows the listed above classification quality metrics depending on values of the coefficient R . In this case, increasing R implies decreasing the amount of noise and, consequently, leads to a complication of classification. Figure 10 shows ROC curves for $R = 1, 1.9, 7$ for this case.

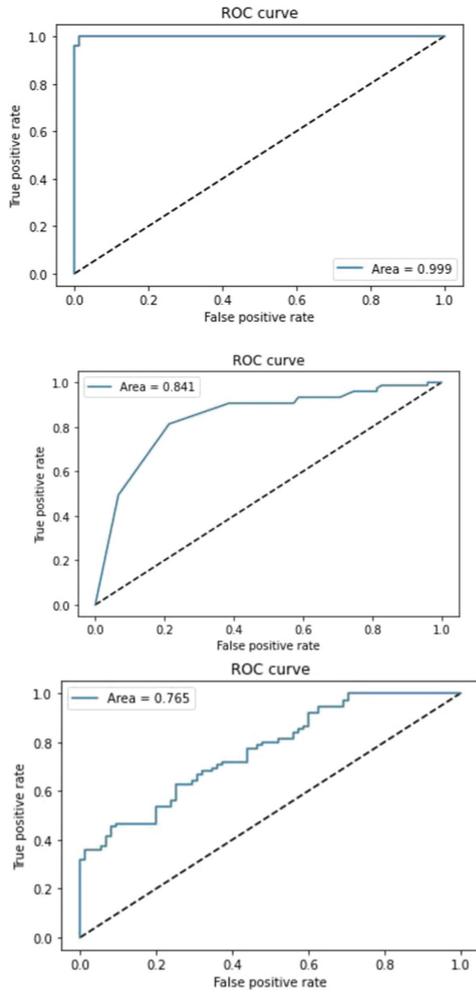


Fig. 10. ROC curves for $R = 1, 1.9, 7$ (from top to bottom)

Table 3 shows the quality metrics depending on values of R . In this case, an increase in R means a decrease in the level of noise and, consequently, results in degradation of classification quality.

Table 3. Classification metrics for clean and noisy TS-samples

R	Accuracy	Precision	Recall	f1 score	ROC AUC
1	0.94	0.85	0.99	0.93	0.999
1.5	0.91	0.91	0.93	0.92	0.99
1.9	0.83	0.87	0.79	0.83	0.841
7.0	0.65	0.62	0.75	0.68	0.765

6 Conclusions

In the work, a classification of time realizations is carried out on the basis of the recognition of the corresponding wavelet spectra. A continuous wavelet transform was applied to the time realizations, which generated a two-dimensional array of values of the wavelet coefficients. The transition from a signal in the time domain to an image of the frequency-time surface made it possible to obtain visual information about the change in time of the relative contribution of components of different scales. The resulting images were classified using the ResNet50 deep neural network. The paper considers two classification problems related to noisy time realizations. For numerical experiments sinusoidal time realizations with and without additive normal noise were chosen. In the first case, the classes are noisy realizations of sinusoids with frequencies from different ranges. In the second case, all time realizations are divided into two classes: noisy and no noise. The obtained results confirm that the classification based on image recognition of wavelet spectra makes it possible to distinguish well signals with an additive noise component with different signal/noise levels. This allows us to recommend the application of the proposed method for classifying noisy time series of different types.

Future research will focus on applying this approach to real datasets and developing a methodology for choosing the optimal wavelet function to visualize the characteristic features of the realization in the frequency and time domains.

Acknowledgements. The work was supported in part by Beethoven Grant No. DFG-NCN 2016/23/G/ST1/04083.

References

1. Understanding ResNet50 architecture. <https://iq.opengenus.org/resnet50-architecture/>
2. ADMIN: A guide for using the wavelet transform in machine learning. <https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/>
3. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Disc.* **31**(3), 606–660 (2016). <https://doi.org/10.1007/s10618-016-0483-9>

4. Chatfield, C.: *The Analysis of Time Series: An Introduction*, 6th edn. Chapman and Hall/CRC, London (2003)
5. Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, IJCAI 2011*, vol. 2, pp. 1237–1242. AAAI Press (2011)
6. Dau, H.A., et al.: The UCR time series archive. [arXiv:1810.07758](https://arxiv.org/abs/1810.07758) [cs, stat] <http://arxiv.org/abs/1810.07758> (2019)
7. Esling, P., Agon, C.: Time-series data mining. *ACM Comput. Surv.* **45**(1), 12:1–12:34 (2012). <https://doi.org/10.1145/2379776.2379788>
8. Faraggi, M., Sayadi, K.: Time series features extraction using fourier and wavelet transforms on ECG data. <https://blog.octo.com/time-series-features-extraction-using-fourier-and-wavelet-transforms-on-ecg-data/>
9. Feike, S.: Multiple time series classification by using continuous wavelet transformation. <https://towardsdatascience.com/multiple-time-series-classification-by-using-continuous-wavelet-transformation-d29df97c0442>
10. Feng, V.: An overview of ResNet and its variants. <https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035>
11. Gamba, J.C.B.: Deep learning for time-series analysis. [arXiv:1701.01887](https://arxiv.org/abs/1701.01887) [cs] <http://arxiv.org/abs/1701.01887> (2017)
12. Hamilton, J.D.: *Time Series Analysis*, 1st edn. Princeton University Press, Princeton (1994)
13. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A.: Deep learning for time series classification: a review. *Data Min. Knowl. Disc.* **33**(4), 917–963 (2019). <https://doi.org/10.1007/s10618-019-00619-1>
14. Ivanisenko, I., Kirichenko, L., Radivilova, T.: Investigation of multifractal properties of additive data stream. In: *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, pp. 305–308 (2016). <https://doi.org/10.1109/DSMP.2016.7583564>
15. Janczura, J., Kowalek, P., Loch-Olszewska, H., Szwabiński, J., Weron, A.: Classification of particle trajectories in living cells: machine learning versus statistical testing hypothesis for fractional anomalous diffusion. *Phys. Rev. E* **102**(3), 032402 (2020). <https://doi.org/10.1103/PhysRevE.102.032402>
16. Kirichenko, L., Radivilova, T.: Analyzes of the distributed system load with multifractal input data flows. In: *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pp. 260–264 (2017). <https://doi.org/10.1109/CADSM.2017.7916130>
17. Kirichenko, L., Radivilova, T., Bulakh, V., Zinchenko, P., Saif Alghawli, A.: Two approaches to machine learning classification of time series based on recurrence plots. In: *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, pp. 84–89 (2020). <https://doi.org/10.1109/DSMP47368.2020.9204021>
18. Kirichenko, L., Zinchenko, P., Radivilova, T.: Classification of time realizations using machine learning recognition of recurrence plots. In: Babichev, S., Lytyvenko, V., Wójcik, W., Vyshemyrskaya, S. (eds.) *ISDMCI 2020*. AISC, vol. 1246, pp. 687–696. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-54215-3_44
19. Li, D., Bissyande, T.F., Klein, J., Traon, Y.L.: Time series classification with discrete wavelet transformed data. *Int. J. Softw. Eng. Knowl. Eng.* **26**(9), 1361–1377 (2016). <https://doi.org/10.1142/S0218194016400088>

20. Långkvist, M., Karlsson, L., Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recogn. Lett.* **42**, 11–24 (2014). <https://doi.org/10.1016/j.patrec.2014.01.008>
21. Maharaj, E.A., D’Urso, P., Caiado, J.: *Time Series Clustering and Classification*, 1st edn. Chapman and Hall/CRC, London (2021)
22. Newland, D.E.: *An Introduction to Random Vibrations, Spectral & Wavelet Analysis*, 3rd edn. Dover Publications, Mineola (2005)
23. Nwe, T.L., Dat, T.H., Ma, B.: Convolutional neural network with multi-task learning scheme for acoustic scene classification. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1347–1350 (2017). <https://doi.org/10.1109/APSIPA.2017.8282241>
24. Nweke, H.F., Teh, Y.W., Al-garadi, M.A., Alo, U.R.: Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges. *Expert Syst. Appl.* **105**, 233–261 (2018). <https://doi.org/10.1016/j.eswa.2018.03.056>
25. Rajkomar, A., et al.: Scalable and accurate deep learning with electronic health records. *npj Dig. Med.* **1**(1), 1–10 (2018). <https://doi.org/10.1038/s41746-018-0029-1>
26. O Sirait, H., et al.: Time frequency signal classification using continuous wavelet transformation. In: *IOP Conference Series: Materials Science and Engineering*, vol. 851, no. 1, p. 012045 (2020). <https://doi.org/10.1088/1757-899X/851/1/012045>
27. Susto, G.A., Cenedese, A., Terzi, M.: Chapter 9 - time-series classification methods: review and applications to power systems data. In: Arghandeh, R., Zhou, Y. (eds.) *Big Data Application in Power Systems*, pp. 179–220. Elsevier (2018). <https://doi.org/10.1016/B978-0-12-811968-6.00009-7>
28. Wickerhauser, M.V.: *Adapted Wavelet Analysis: From Theory to Software*, 1st edn. A K Peters/CRC Press, Natick (2019)
29. Yang, Q., Wu, X.: 10 challenging problems in data mining research. *Int. J. Inf. Technol. Decis. Mak.* **05**(4), 597–604 (2006). <https://doi.org/10.1142/S0219622006002258>
30. Zhang, H., Ho, T.B., Lin, M.-S., Liang, X.: Feature extraction for time series classification using discriminating wavelet coefficients. In: Wang, J., Yi, Z., Zurada, J.M., Lu, B.-L., Yin, H. (eds.) *ISNN 2006. LNCS*, vol. 3971, pp. 1394–1399. Springer, Heidelberg (2006). https://doi.org/10.1007/11759966_207



A Noise Resistant Credibilistic Fuzzy Clustering Algorithm on a Unit Hypersphere with Illustrations Using Expression Data

Zhengbing Hu¹, Mark Last², Tzung-Pei Hong^{3,4},
Oleksii K. Tyshchenko⁵, and Esha Kashyap⁶

¹ School of Computer Science, Hubei University of Technology, Wuhan, China

² Information Systems Engineering and Software Engineering, Ben-Gurion University of the Negev, Be'er Sheva, Israel

mlast@bgu.ac.il

³ National University of Kaohsiung, Kaohsiung, Taiwan

tphong@nuk.edu.tw

⁴ National Sun Yat-sen University, Kaohsiung, Taiwan

⁵ Institute for Research and Applications of Fuzzy Modeling, CE IT4Innovations, University of Ostrava, Ostrava, Czech Republic

lehatish@gmail.com

⁶ Ramanujan School of Mathematical Sciences, Pondicherry University, Puducherry, India

Abstract. This article presents a robust noise-resistant fuzzy-based algorithm for cancer class detection. High-throughput microarray technologies facilitate the generation of large-scale expression data; this data captures enough information to build classifiers to understand the molecular basis of a disease. The proposed approach built on the Credibilistic Fuzzy C-Means (CFCM) algorithm partitions data restricted to a p -dimensional unit hypersphere. CFCM was introduced to address the noise sensitiveness of fuzzy-based procedures, but it is unstable and fails to capture local non-linear interactions. The introduced approach addresses these shortcomings. The experimental findings in this article focus on cancer expression datasets. The performance of the proposed approach is assessed with both internal and external measures. The fuzzy-based learning algorithms Fuzzy C-Means (FCM) and Hyperspherical Fuzzy C-Means (HFCM) are used for comparative analysis. The experimental findings indicate that the proposed approach can be used as a plausible tool for clustering cancer expression data.

Keywords: Fuzzy clustering · Credibilistic fuzzy c-means · Spherical space · Gene expression · Cancer data

1 Introduction

High-throughput microarray technologies facilitate the generation of large-scale expression data. These techniques record the molecular signatures of disease;

molecular measurements are used to build classifiers to predict disease groups. Large-scale expression data is associated with a humongous portion of molecular features, but a minuscule of these features are germane to clinical diagnosis and disease prediction. There is no information on the association of these features to a diseased state [31]. The molecular signature of different disease groups differs more in direction [2]. Furthermore, the patient cohort in expression data is infinitesimal in comparison to these molecular features. This results in the 'short-fat data' problem; the feature space becomes sparse in a geometrical sense.

In the past decades, a multitude of data clustering approaches was focused on this bio-molecular data; cancer detection focused on this data provides better insight into cancer groups. Data clustering approaches partition data into groups based either on a distance or a correlation measure, or a statistical data distribution of the space. Fuzzy-based data clustering approaches use fuzzy mathematics with crisp clustering. These algorithms obtain a soft partition of the data, a data point is assigned to multiple clusters with membership in $[0,1]$. These algorithms furnish clinicians with a better linguistic understanding of the data [23]. These approaches are self-learning [14], independent of training samples, and robust to handle imprecise information associated with data features [1, 20, 31].

The main contribution of the paper can be formulated in the following manner:

1. A fuzzy-based learning algorithm Credibilistic Fuzzy C-Means-Direction (CFCM-D) focusing on directional characteristics of the expression data is introduced.
2. CFCM-D is based on the Credibilistic Fuzzy C-Means (CFCM) algorithm and partitions multivariate data restricted to a p -dimensional hypersphere of a unit radius; this restriction construes feature vectors as directional data.
3. The experimental comparison of the proposed approach with FCM and HFCM was based on nine expression datasets.

CFCM algorithm was introduced to reduce the noise sensitiveness of the standard Fuzzy C-Means (FCM) algorithm. It retains some of the basic limitations of FCM; analogous to FCM, CFCM fails to capture local non-linear interactions in the data [18, 32] and detect non-globular cluster structures. It assigns outliers to multiple clusters [17, 18]. Additionally, the CFCM algorithm is unstable; it uses a measure ψ_i estimated with unstable cluster centers undergoing a convergence process. CFCM-D uses a cosine distance measure; the cosine distance captures an angular separation between data points and is independent of the length of patterns under consideration. The robustness of CFCM-D depends on two measures, ψ_i , and w_i . ψ_i assists in the search of outliers in the data. ψ_i in CFCM-D accounts for the data distribution in space and is independent of the cluster centers. ψ_i depends on the cosine distance between the data points and the geometric median of the data space. In a mathematical sense, the distance between an outlier and the geometric median should be minimum, thereupon outliers are assigned minimum ψ_i 's. w_i presents local information on data points. It searches for local dense neighborhoods in the data space accounting for the data distribution in space. It is determined by the RBF kernel function.

The remainder of this manuscript is structured as follows. Section 2 presents the past research on data clustering in a spherical space. Section 3 provides a brief introduction to three fuzzy-based clustering algorithms FCM, Hyperspherical Fuzzy C-Means (HFCM), and CFCM. Section 4 furnishes the theoretical formulation of the proposed approach (CFCM-D). Section 5 discusses the offered approach's performance using nine cancer expression datasets. Section 6 concludes this article.

2 Related Works

In literature, numerous research papers are focused on unsupervised data partitioning in the spherical space or partitioning data restricted to the spherical space. These methods can be fractionated into distribution-based approaches and distribution-free approaches; each approach has limitations of its own. This section provides a brief of these approaches proposed in the recent past.

The authors in [16] proposed an extension of the standard K-Means algorithm, the `spkmeans` algorithm. `spkmeans` was focused on large-scale sparse high-dimensional data. `spkmeans` clustered data using a cosine distance measure. In [3], the authors presented three data partitioning algorithms based on `spkmeans`: `fs-spkmeans`, `ofs-spkmeans` and `fscl-spkmeans` to group high-dimensional data on a hypersphere of unit radius with frequency sensitive assignments. The standard K-Means and `spkmeans` algorithms get jammed to local solutions and produce imbalanced cluster structures for high-dimensional data; these three algorithms were an attempt to resolve this issues with frequency sensitive competitive learning (FSCL). In [4], three FSCL algorithms were proposed to search for balanced clusters in high-dimensional data: `pifs-spkmeans` and `fifs-spkmeans` were proposed to partition static data and `sfs-spkmeans` for streaming data. The authors used von-Mises Fisher (vMF) distribution to model directional data. In [3] and [4], experiments focused on high-dimensional text documents. The authors in [2] presented a generative model to partition directional data distributed on a hypersphere of unit radius. The generative mixture-model was based on von-Mises Fisher distribution. The authors proposed two learning algorithms `hard-movMF` and `soft-movMF` based on the Expectation Maximization (EM) algorithm to partition directional data. `Hard-movMF` generated a crisp partition of the data while `soft-movMF` generated a soft partition. The experimental investigation focused on a benchmark with text documents and expression datasets. The robustness of the proposed algorithms was compared with `spkmeans` and `fs-spkmeans` algorithms. `Soft-movMF` demonstrated the superior performance. In [12], the authors proposed a model-based EM learning algorithm to cluster multivariate yeast expression data on a sphere of radius one. The data on the sphere was clustered with inverse stereographic projections of multivariate Gaussian distribution. The model-based algorithm showed a performance gain in comparison with the K-Means and MClust algorithms. K-mean-directions algorithm was proposed in [21]. The algorithm was based on the generic partition K-Means algorithm. In this proposed scheme, data was restricted to a

p-dimensional hypersphere of radius one; the correlation distance measure was used. The performance of the K-mean-directions algorithm was assessed with a series of large-scale simulated experiments, time-course yeast expression data, and text documents. The experimental investigation found K-mean-directions to be faster in comparison with the K-Means and spkmeans algorithms.

In [28], the authors presented an unsupervised algorithm to cluster directional data constrained to a unit hypersphere based on Similarity-based Clustering Method (SCM) [30]. The algorithm was found to be robust to initialization and noise in the data. The proposed algorithm was used to group the data with mixtures of the Fisher's distribution and exoplanet dataset. Fuzzy-based algorithm focused on directional data was foremost presented in [29]. The authors proposed the Fuzzy C-Direction (FCD) algorithm; FCD was based on the FCM algorithm. It clustered directional data modeled after the vMF distribution. The authors in [22] proposed a fuzzy-based algorithm HFCM for grouping high-dimensional sparse directional data. HFCM is also based on the FCM algorithm with an analogous objective function. It uses the cosine distance measure. The authors assessed the robustness of the algorithm with the traditional K-Means. Text documents were used in the experiment. HFCM exhibited superior performance; precision and recall were used to assess the algorithm's performance. In [27], Hyperspherical Possibilistic Fuzzy C-Means (HPFCM) algorithm focused on document clustering was proposed. Partitioning occurred on a p-dimensional hypersphere of unit radius. HPFCM was based on the Possibilistic Fuzzy C-Means (PFCM) algorithm. HPFCM partitioned the document data with the cosine distance measure. The robustness of HPFCM was assessed with standard document datasets. HPFCM showed decisive performance gains in comparison with PFCM and HFCM algorithms. HPFCM assisted in global optimization. However, HPFCM is dependent on multiple parameters, making it difficult for selection of an optimal parameter and it was found sensitive to initialization. In [19], the authors presented a distribution-free learning algorithm to cluster directional data. The proposed algorithm was based on FCM. The algorithm was named FCM4DD. FCM4DD partitioned circular data in $[-\pi, \pi)$ or $[0, 2\pi)$. The angular separation was used as a distance measure. The efficiency of the proposed algorithm was investigated by four artificial directional datasets and one real-world turtle direction dataset; the comparative investigation was carried out by FCD and EM algorithms. The FCM4DD partitioned data with the directional distribution and was found less sensitive to initialization. The authors in [5] proposed the Possibilistic C-Means Direction (PCM-D) algorithm for directional data. PCM-D was proposed as an extension to the Possibilistic C-Means (PCM) algorithm on the directional data. It partitioned data on an unit p-dimensional hypersphere with the cosine distance measure. The experimental findings were focused on artificial datasets generated by mixtures of hyperspherical distribution on the unit hypersphere and applied for real-world animal orientation and wind direction datasets.

3 Background

Clustering approaches search for hidden structures in the data; they partition unlabeled data. A clustering algorithm partitions a dataset, $\{x_i\}_{i=1}^n, x_i \in \mathbb{R}^p$ into c clusters. Fuzzy-based partitioning groups the feature vectors of the dataset into non-crisp clusters; each feature vector x_i is associated with a membership degree in $[0, 1]$. Numerous contribution focused on the fuzzy-based partitioning is present in literature. In this section, three fuzzy-based algorithms FCM, HFCM and CFCM are discussed; a brief introduction is provided.

3.1 Fuzzy C-Means (FCM)

In literature, fuzzy partitioning was first introduced in 1973 [13]. The authors in [6] furthered the idea of fuzzy partitions and proposed the FCM algorithm. FCM is a probabilistic distribution-free algorithm. It searches for soft partitions in the data. It is based on the minimization of the following functional:

$$J_{FCM}(U, V; X) = \sum_{i=1}^n \sum_{k=1}^c u_{ik}^m \|x_i - v_k\|_A^2, \tag{1}$$

with

$$\sum_{k=1}^c u_{ik} = 1, u_{ik} \in [0, 1], \forall i = 1, 2, \dots, n,$$

in which $U = \{u_{ik} | u_{ik} \in [0, 1], \sum_{k=1}^c u_{ik} = 1, \forall i\}$, u_{ik} represents the membership assignment of the i^{th} feature vector x_i to the k^{th} cluster. $V = \{v_k\}_{k=1}^c$ is a set of cluster centers, $v_k \in \mathbb{R}^p$. $m \in (1, \infty]$ is a fuzzy parameter, it controls fuzziness of the membership assignments. $\|x\|_A = \sqrt{x^T A x}$ is an inner product. In general, FCM uses the Euclidean distance measure. Iterative minimization of (1) results in the following membership assignments and cluster centers:

$$u_{ik} = \frac{1}{\sum_{r=1}^c \left\{ \frac{\|x_i - v_k\|_A^2}{\|x_i - v_r\|_A^2} \right\}^{\frac{m}{m-1}}}, \forall i = 1, 2, \dots, n, \forall k = 1, 2, \dots, c,$$

$$v_k = \frac{\sum_{i=1}^n u_{ik}^m x_i}{\sum_{i=1}^n u_{ik}^m}, \forall k = 1, 2, \dots, c.$$

The FCM algorithm fails to detect non-globular clusters in the data. It is sensitive to the initialization of cluster centers and gets stuck to local solutions. FCM produces incorrect memberships in presence of noise and outliers in the data. Furthermore, for high-dimensional data, the feature space is sparse in a geometrical sense and FCM is highly sensitive to the distance measure in use.

3.2 Hyperspherical Fuzzy C-Means (HFCM)

The HFCM algorithm was proposed in [22]. This is one of the foremost distribution-free fuzzy-based algorithm presented for partitioning high-dimensional directional data. In particular, the authors investigated high-dimensional sparse documents to examine document relation. HFCM is based on

the FCM algorithm with analogous objective. It uses the cosine distance measure, focusing on the angular separation of feature vectors. HFCM minimizes the following objective function:

$$J_{HFCM}(U, V; X) = \sum_{i=1}^n \sum_{k=1}^c u_{ik}^m \left\{ 1 - \sum_{j=1}^p x_{ij} v_{kj} \right\}, \tag{2}$$

with

$$\sum_{k=1}^c u_{ik} = 1, u_{ik} \in [0, 1], \forall i = 1, 2, \dots, n,$$

where $X = \{x_1, x_2, \dots, x_n\}$ is a set of p -dimensional documents, $V = \{v_1, v_2, \dots, v_c\}$ is a set of c cluster centers, u_{ik} are fuzzy membership assignments, $\sum_{i=1}^n u_{ik} = 1$ and $0 \leq u_{ik} \leq 1$. $m > 1$ is a fuzzy parameter. $1 - \sum_{j=1}^p x_{ij} v_{kj}$ is the cosine distance function. Iterative minimisation of (2) results in the following membership assignments and cluster centers:

$$u_{ik} = \frac{1}{\sum_{r=1}^c \left\{ \frac{1 - \sum_{j=1}^p x_{ij} v_{kj}}{1 - \sum_{j=1}^p x_{ij} v_{rj}} \right\}^{\frac{1}{m-1}}}, \forall i = 1, 2, \dots, n, \forall k = 1, 2, \dots, c,$$

$$v_k = \sum_{i=1}^n u_{ik}^m x_i \left\{ \sum_{j=1}^p \left\{ \sum_{i=1}^n u_{ik}^m x_{ij} \right\}^2 \right\}^{-\frac{1}{2}}, \forall k = 1, 2, \dots, c.$$

HFCM retains most limitations of the FCM algorithm.

3.3 Credibilistic Fuzzy C-Means (CFCM)

FCM assigns outliers with high membership values; this affects its robustness. The CFCM algorithm was proposed in [10] to address this noise sensitiveness by FCM. CFCM is based on FCM with analogous objective function. In CFCM, the probabilistic constraint $\sum_{k=1}^c u_{ik} = 1$ of FCM is substituted by $\sum_{k=1}^c u_{ik} = \psi_i$, where ψ_i represents the typicality of x_i to an entire dataset. It is based on the minimization of the following functional:

$$J_{CFCM}(U, V; X) = \sum_{i=1}^n \sum_{k=1}^c u_{ik}^m \|x_i - v_k\|_A^2, \tag{3}$$

with

$$\sum_{k=1}^c u_{ik} = \psi_i, u_{ik} \in [0, 1], \forall i = 1, 2, \dots, n,$$

where $U = \{u_{ik} | u_{ik} \in [0, 1], \sum_{k=1}^c u_{ik} = \psi_i, \forall i\}$, u_{ik} represents the membership assignment of the i^{th} feature vector, x_i to the k^{th} cluster. $V = \{v_k\}_{k=1}^c$ is a set of cluster centers, $v_k \in \mathbb{R}^p$. $m \in (1, \infty]$ is a fuzzy parameter, it controls fuzziness of the membership assignments. $\psi_i = f(U, V, X)$. $\|x\|_A = \sqrt{x^T A x}$ is an inner product. The authors provided a further modification of CFCM in [9]. The authors in [9, 10] proposed three formulations of ψ_i :

1.

$$\psi_i = 1 - \frac{\{1 - \theta\}\alpha_i}{\max\{\alpha_1, \alpha_2, \dots, \alpha_n\}}$$

where α_i is a distance of x_i to the nearest cluster center.

2.

$$\psi_i = 1 - \frac{\kappa_i - \min\{\kappa_1, \kappa_2, \dots, \kappa_n\}}{\max\{\kappa_1, \kappa_2, \dots, \kappa_n\} - \min\{\kappa_1, \kappa_2, \dots, \kappa_n\}}, i = 1, 2, \dots, n,$$

where κ_i is the mean distance of x_i from its ς nearest neighbours; $\kappa_i = \frac{\sum_{j=1}^{\varsigma} (\|y_i^j - x_i\|)}{\varsigma}$.

3.

$$\psi_i = \exp(-(\frac{\kappa_i}{\iota\omega})^2)$$

where ι is a user-defined parameter; $\iota = \sqrt{\frac{-1}{\log \tau}}, \tau \in (0, 1)$ and $\omega^2 = \{\frac{1}{n} \{\sum_{i=1}^n \kappa_i^{-2}\}\}^{-1}$ is the harmonic second moment of the data.

Minimization of (3) results in the following membership assignments and cluster centers:

$$u_{ik} = \frac{\psi_i}{\sum_{r=1}^c \{ \frac{\|x_i - v_k\|_A^2}{\|x_i - v_r\|_A^2} \}^{\frac{2}{m-1}}}, \forall i = 1, 2, \dots, n, \forall k = 1, 2, \dots, c,$$

$$v_k = \frac{\sum_{i=1}^n u_{ik}^m x_i}{\sum_{i=1}^n u_{ik}^m}, \forall k = 1, 2, \dots, c.$$

The original CFM algorithm is unstable; it uses a measure ψ_i estimated by unstable cluster centers undergoing the convergence process. CFM assigns minimum memberships to outliers in the data; apart from this, it retains most limitations of the standard FCM procedure.

4 Credibilistic Fuzzy C-Means-Direction (CFM-D)

This section discusses the theoretical formulation of the proposed approach, CFM-D. A robust fuzzy clustering algorithm based on the CFM approach is presented. The main motive is to build a noise-resistant fuzzy-based learning procedure to cluster the directional data. This section is fractionated into two subsections: Subject. 4.1 discusses a standardization methodology for multivariate feature vectors onto a hypersphere of unit radius, the clustering process occurs in this spherical space; Subject. 4.2 describes the proposed approach.

4.1 Standardization of the Feature Vectors on a Unit Hypersphere

Consider a dataset X with multivariate feature vectors in \mathbb{R}^p : $X = \{x_i\}_{i=1}^n, x_i \in \mathbb{R}^p$. The i^{th} feature vector x_i is a p-tuple of the form:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ip}).$$

The mean \bar{x}_i and standard deviation σ_{x_i} of x_i are defined as:

$$\bar{x}_i = \frac{\sum_{j=1}^p x_{ij}}{p}; \forall i = 1, 2, \dots, n$$

and

$$\sigma_{x_i} = \left\{ \frac{\sum_{j=1}^p \{x_{ij} - \bar{x}_i\}^2}{p} \right\}^{\frac{1}{2}}, \forall i = 1, 2, \dots, n.$$

The coordinates of the multivariate feature vector $x_i, i = 1, 2, \dots, n$ are standardized by:

$$\tilde{x}_{ij} = \sqrt{\frac{1}{p}} \cdot \frac{x_{ij} - \bar{x}_i}{\sigma_{x_i}}, \forall j = 1, 2, \dots, p,$$

in which $\tilde{x}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, \dots, \tilde{x}_{ip}), i = 1, 2, \dots, n$ is the standardized feature vector. This construction of \tilde{x}_{ij} , allows the standardized feature vectors \tilde{x}_i to satisfy the following:

$$\sum_{j=1}^p \tilde{x}_{ij} = 0; \sum_{j=1}^p \tilde{x}_{ij}^2 = 1, \forall i = 1, 2, \dots, n$$

That is, a standardized feature vector \tilde{x}_i lies at the intersection of a hyperplane and a unit hypersphere in \mathbb{R}^p . This data standardization process is discussed in [12]. The proposed approach clusters this standardized data, $\tilde{X} = \{\tilde{x}_i\}_{i=1}^n$ on a unit hypersphere in \mathbb{R}^p .

4.2 Method

The clustering process is based on the constrained minimization of the following functional:

$$J_{CFCM-D}(U, V; \tilde{X}) = \sum_{i=1}^n \sum_{k=1}^c w_i u_{ik}^m \left\{ 1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj} \right\} \tag{4}$$

with

$$\sum_{k=1}^c u_{ik} = \psi_i, u_{ik} \in [0, 1], \forall i = 1, 2, \dots, n, \tag{5}$$

where $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$ is the standardized p-dimensional dataset on a hypersphere of unit radius in \mathbb{R}^p . $V = \{v_k\}_{k=1}^c$, v_k is the cluster center of the k^{th} cluster; $U = [u_{ik}]_{n \times c}$, u_{ik} is the membership assignment of the i^{th} feature vector to the k^{th} cluster. v_k also lies on the p-dimensional unit hypersphere; $\sum_{j=1}^p v_{kj}^2 = 1$. $\sum_{i=1}^n \sum_{k=1}^c w_i u_{ik}^m \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}\}$ in (4) measures the distortion between the features vectors and the cluster centers, in which w_i are generated with RBF kernel function. ψ_i in (5) is determined prior to the clustering process; it estimates the typicality of the i^{th} feature vector \tilde{x}_i to the dataset. In the proposed scheme, ψ_i is a function of \tilde{x}_i , $\psi_i = f(\tilde{x}_i, \varrho)$, $\varrho = \arg \min_{u \in \mathbb{R}^p} \sum_{i=1}^n \{1 - \sum_{j=1}^p \tilde{x}_{ij} u_j\}$

is the geometric median of \tilde{X} . ϱ is a robust central representative of data with noise. ψ_i is defined as follows:

$$\psi_i = 1 - \frac{\{1 - \vartheta\}\kappa_i}{\max_{r=1,2,\dots,n} \kappa_r}, \tag{6}$$

in which $\kappa_i = \{1 - \sum_{j=1}^p \tilde{x}_{ij}\varrho_j\}$. κ_i measures the cosine distance between the i^{th} feature vector \tilde{x}_i and the geometric median ϱ . ϑ ascertains the minimum of ψ_i . $0 < \vartheta < 1$. Further, $\psi_l = \vartheta$, when $\kappa_l = \max_{r=1,2,\dots,n} \kappa_r$. w_i is defined as follows:

$$w_i = \begin{cases} 1, & \sum_{k=1}^n \exp(-\frac{\{1 - \sum_{j=1}^p \tilde{x}_{ij}\tilde{x}_{kj}\}}{\min_{r=1,2,\dots,n} \kappa_r})^2 > 0.1, \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

w_i 's are modeled after the RBF kernel function described in [24]. w_i of \tilde{x}_i corresponds to $|\tilde{x}_l|_{l \neq i}^n$ closer than $\min_{r=1,2,\dots,n} \kappa_r$ to \tilde{x}_i ; w_i assures cluster centers are less sensitive to the noise in the data. The proposed algorithm is solved by differentiating the Lagrangian function:

$$\begin{aligned} L(U, V, \beta_i, \xi_k; \tilde{X}) &= J(U, V; \tilde{X}) + \sum_{i=1}^n \beta_i \sum_{k=1}^c \{\psi_i - u_{ik}\} + \sum_{k=1}^c \xi_k \left\{ \sum_{j=1}^p v_{kj}^2 - 1 \right\} \\ &= \sum_{i=1}^n \sum_{k=1}^c w_i u_{ik}^m \left\{ 1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj} \right\} + \sum_{i=1}^n \beta_i \sum_{k=1}^c \{\psi_i - u_{ik}\} + \sum_{k=1}^c \xi_k \left\{ \sum_{j=1}^p v_{kj}^2 - 1 \right\} \end{aligned} \tag{8}$$

with respect to u_{ik} and v_k . This transforms the minimization problem to an unconstrained problem. β_i and ξ_k are the Lagrange multipliers. Differentiating $L(U, V, \beta_i, \xi_k; \tilde{X})$ with respect to u_{ik} :

$$\begin{aligned} \frac{\partial L(U, V, \beta_i, \xi_k; \tilde{X})}{\partial u_{ik}} &= 0 \\ \text{or, } mw_i u_{ik}^{m-1} \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}\} - \beta_i &= 0 \\ \text{or, } mw_i u_{ik}^{m-1} \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}\} &= \beta_i. \\ \text{or, } u_{ik}^{m-1} &= \frac{\beta_i}{mw_i \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}\}} \end{aligned}$$

Thus,

$$u_{ik} = \left\{ \frac{\beta_i}{mw_i \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}\}} \right\}^{\frac{1}{m-1}} \tag{9}$$

substituting this in (5),

$$\begin{aligned} \sum_{k=1}^c \left\{ \frac{\beta_i}{mw_i \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}\}} \right\}^{\frac{1}{m-1}} &= \psi_i \\ \text{or, } \beta_i \sum_{k=1}^c \frac{1}{mw_i \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}\}} &= \psi_i^{m-1} \\ \text{or, } \beta_i = \psi_i^{m-1} \sum_{r=1}^c mw_i \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{rj}\}. \end{aligned}$$

Replacing β_i in (9) leads to:

$$u_{ik} = \frac{\psi_i}{\left\{ \frac{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{kj}}{\sum_{r=1}^c \{1 - \sum_{j=1}^p \tilde{x}_{ij} v_{rj}\}} \right\}^{\frac{1}{m-1}}}, i = 1, 2, \dots, n, k = 1, 2, \dots, c. \tag{10}$$

Differentiating $L(U, V, \beta_i, \xi_k; \tilde{X})$ with respect to v_k :

$$\begin{aligned} \frac{\partial L(U, V, \beta_i, \xi_k; \tilde{X})}{\partial v_k} &= 0 \\ \text{or, } -\sum_{i=1}^n w_i u_{ik}^m \tilde{x}_i + 2\xi_k v_k &= 0 \\ \text{or, } 2\xi_k v_k &= \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_i. \text{ Thus,} \end{aligned}$$

$$v_k = \frac{1}{2\xi_k} \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_i. \tag{11}$$

v_k lies on the p-dimensional hypersphere in \mathbb{R}^p , so

$$\begin{aligned} \sum_{j=1}^p v_{kj}^2 &= 1 \\ \text{or, } \sum_{j=1}^p \left\{ \frac{1}{2\xi_k} \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_{ij} \right\}^2 &= 1 \\ \text{or, } \frac{1}{4\xi_k^2} \sum_{j=1}^p \left\{ \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_{ij} \right\}^2 &= 1 \\ \text{or, } \frac{1}{4\xi_k^2} &= \frac{1}{\sum_{j=1}^p \left\{ \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_{ij} \right\}^2} \\ \text{or, } \frac{1}{2\xi_k} &= \pm \left\{ \frac{1}{\sum_{j=1}^p \left\{ \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_{ij} \right\}^2} \right\}^{\frac{1}{2}}, \text{ since } \xi_k > 0: \\ \frac{1}{2\xi_k} &= \left\{ \sum_{j=1}^p \left\{ \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_{ij} \right\}^2 \right\}^{-\frac{1}{2}}. \text{ Thus, the updated cluster center } v_k \text{ is as follows:} \end{aligned}$$

$$v_k = \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_i \left\{ \sum_{j=1}^p \left\{ \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_{ij} \right\}^2 \right\}^{-\frac{1}{2}}, 1 \leq k \leq c. \tag{12}$$

5 Performance Metrics

The robustness of different clustering approaches is assessed with internal and external measures. Internal measures are performance metrics based on information within the data; external measures are based on external information. The experimental investigation uses five performance metrics. These measures are discussed in this section.

5.1 Precision and Recall

Precision and recall measure the success of prediction of a classifier. These measures are used in unison. Precision measures the accurateness in positive prediction of a classifier. Recall measures the accurateness in prediction of a classifier. Precision and recall are defined as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}},$$

positive-negative refers to the classifier prediction and true-false ascertains whether the prediction is in accordance with the ground truth. Precision and recall are external measures. Precision $\in [0, 1]$. Recall $\in [0, 1]$.

5.2 Silhouette Coefficient (SC)

For c partition $\{C_k\}_{k=1}^c$ of $\{x_i\}_{i=1}^n$, SC [26] is defined as follows:

$$\text{SC} = \frac{1}{c} \sum_{k=1}^c \frac{1}{|C_k|} \sum_{x \in C_k} \frac{\left\{ \min_{\substack{i=1,2,\dots,c \\ i \neq k}} \frac{1}{|C_i| \sum_{x \in C_i} d(x, x_i)} - \frac{1}{|C_k|-1} \sum_{x_k \in C_k} d(x, x_k) \right\}}{\max \left\{ \frac{1}{|C_k|-1} \sum_{x_k \in C_k} d(x, x_k), \min_{\substack{i=1,2,\dots,c \\ i \neq k}} \frac{1}{|C_i| \sum_{x \in C_i} d(x, x_i)} \right\}},$$

in which $\frac{1}{|C_k|-1} \sum_{x_k \in C_k} d(x, x_k)$ measures the intra-cluster distance and $\min_{\substack{i=1,2,\dots,c \\ i \neq k}} \frac{1}{|C_i| \sum_{x \in C_i} d(x, x_i)}$ the inter-cluster distance. A robust algorithm maximizes the SC score. $\text{SC} \in [-1, 1]$.

5.3 Davies-Bouldin (DB) Score

The DB score [11] is an internal measure. The DB score is measured as follows:

$$\bar{R} = \frac{1}{c} \sum_{i=1}^c R_i$$

where $R_i = \max R_{ij}, R_{ij} = \frac{S_i + S_j}{M_{ij}}$; M_{ij} measures the separation between the clusters i and j and S_i measures the intra-cluster dispersion in the cluster i . A robust algorithm minimizes the DB score. $\text{DB} \in [-\infty, \infty]$.

5.4 Calinski-Harabasz (CH) Score

The CH score [7] is computed as a ratio of two matrices: a between-cluster scatter matrix (BCSM) and a within-cluster scatter matrix (WCSM). The CH score is defined as follows:

$$\text{CH} = \frac{\text{BCSM}}{c-1} \cdot \frac{n-c}{\text{WCSM}}.$$

A cohesive cluster corresponds to the maximization of BCSM and minimization of WCSM. Thereupon, optimal partitioning occurs at the maximum CH score. CH score is unbounded, $\text{CH} \in [-\infty, \infty]$.

6 Results and Discussion

This section presents an experimental assessment of the proposed approach, CFCM-D. The experiments focus on the genomic data. In order to assess the robustness of CFCM-D, a series of experiments with CFCM-D, the standard FCM, and HFCM is carried out. A motive of this series of experiments is a thorough comparison of the performance of these three algorithms.

The experimentation in this article examines nine cancer expression datasets with GSE accession: GSE200894, GSE116312, GSE97332, GSE159518, GSE33126, GSE185770, GSE168466, and GSE50161. Table 1 records the data features. These datasets are procured from the GEO database of NCBI and are associated with external information.

Table 1. Data

GSE	Dimension	Class	Class distribution
GSE200894	6×21447	2	3:3
GSE116312	13×21525	3	3:3:7
GSE97332	14×3480	2	7:7
GSE159518	16×25271	2	6:10
GSE33126	18×48802	2	9:9
GSE185770	18×135750	2	9:9
GSE168466	40×58056	4	10:10:10:10
GSE14520	41×22268	2	19:22
GSE50161	130×21050	5	13:15:22:34:46

1. **GSE200894** records the expression of human lung adenocarcinoma patients. In this microarray experiment, the lung adenocarcinomas were tolerant to a anti-neoplastic drug; the experiment aimed to investigate the biological mechanism behind the tolerance. GSE200894 is with three normal and three osimertinib-resistant PC-9 lung adenocarcinoma cell line. Dimension: 6×21447. Class: 2. Class distribution: 3:3.
2. **GSE116312** records the expression changes in human gastric tissues with three gastric cancer tissues, three CAG tissues and seven FG tissues. Dimension: 13×21525. Class: 3. Class distribution: 3:3:7.
3. **GSE97332** records the circRNA signatures in paired human HCC tissues with seven cancerous and seven non-cancerous tissues. Dimension: 14×3480. Class: 2. Class distribution: 7:7.
4. **GSE159518** records the abnormal expression changes in M.musculus HCC tissues with ten cancerous and six non-cancerous tissues. Dimension: 16×25271. Class: 2. Class distribution: 6:10.
5. **GSE33126** [8] record the microRNA signatures of paired human colon tissue with nine cancerous and nine non-cancerous tissue. Dimension: 18×48802. Class: 2. Class distribution: 9:9.

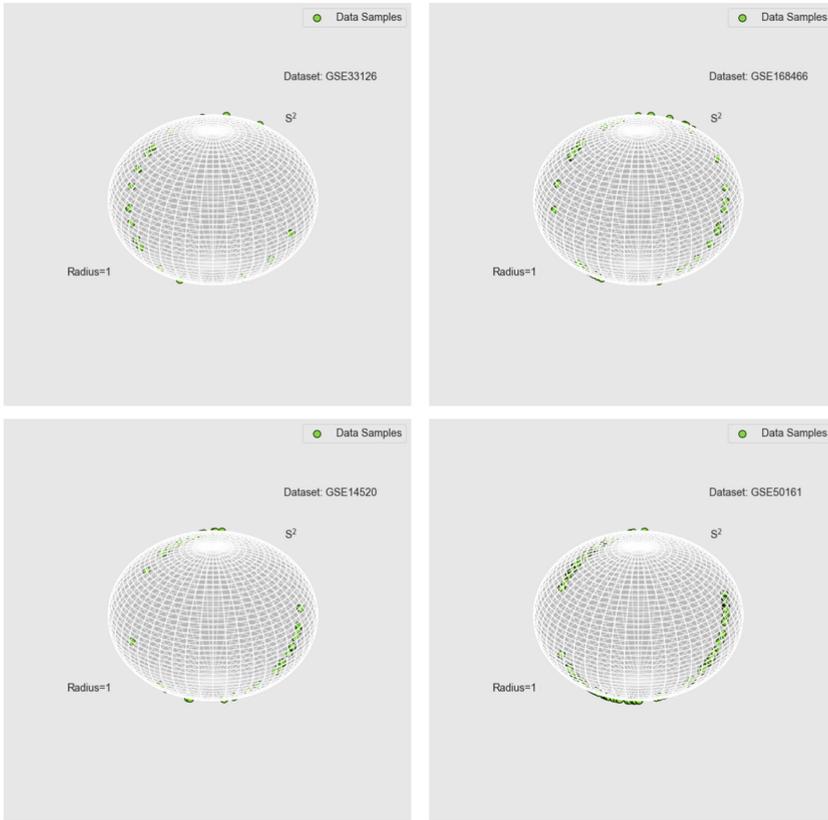


Fig. 1. Data distribution in \mathbb{R}^3 : Data $\{\tilde{x}_i\}_{i=1}^n$ on the intersection of $\sum_{j=1}^3 \tilde{x}_{ij} = 0$ and $\sum_{j=1}^3 \tilde{x}_{ij}^2 = 1$

Table 2. Experimental parameters m and ϑ for CFCM-D

GSE	m	ϑ
GSE200894	2	0.5
GSE116312	2	0.5
GSE97332	2	0.5
GSE159518	2	0.5
GSE33126	2	0.4
GSE185770	1.1	0.1
GSE168466	2	0.5
GSE14520	2	0.5
GSE50161	2	0.2

6. **GSE185770** records the altercations in expression of abnormal colonic mucosa; the expression changes in paired human colorectal tissue with nine cancerous and nine non-cancerous tissue is recorded. Dimension: 18×135750 . Class: 2. Class distribution: 9:9.
7. **GSE168466** records the expression of 98 human NSCLC specimens; a subseries of 40 bio-specimens is used here for experimentation. This bio-specimens are fractionated into four groups: 20 paired BAC and 20 paired AC. Dimension: 40×58056 . Class: 4. Class distribution: 10:10:10:10.
8. **GSE14520** [25] records the abnormal expression changes in 22 paired HCC tissues; here the expression changes of 22 cancerous and 19 non-cancerous HCC tissues is considered. Dimension: 41×22268 . Class: 2. Class distribution: 19:22.
9. **GSE50161** [15] records the expression changes in normal and surgical brain tumor patients. The dataset has 15 normal brain tissues and 115 surgical brain tumor tissues. The 115 surgical brain tumor tissues are further fractionated into four disease groups: ependymoma (46), glioblastoma (34), medulloblastoma (22) and pilocytic astrocytoma (13). Dimension: 130×21050 . Class: 5. Class distribution: 13:15:22:34:46.

CAG: chronic atrophic gastritis, FG: follicular gastritis, HCC: hepatocellular carcinoma, NSCLC: non-small cell lung cancer, BAC: bronchoalveolar carcinoma, AC: alveolar carcinoma.

Table 3. $[\min u_{ik}, \max u_{ik}]$ for CFCM-D

GSE	Range
GSE200894	[0.010161, 0.806320]
GSE116312	[0.000092, 0.934575]
GSE97332	[0.008831, 0.646683]
GSE159518	[0.005391, 0.689870]
GSE33126	[0.018164, 0.852236]
GSE185770	[0.015484, 0.796871]
GSE168466	[0.000097, 0.608135]
GSE14520	[0.000563, 0.642884]
GSE50161	[0.000021, 0.607929]

In microarray experiments, the patient cohort is small but associated with tens of thousands of attributes. This results in the 'short-fat data' problem. The datasets recorded in Table 1 show this characteristic. A reduced low-dimensional data representation in the high-dimensional space is enough to preserve inherent data characteristics. Dimension reduction for high-dimensional expression data is attained using the basic notions of linear algebra and statistics. Principal Component Analysis (PCA) is used for linear dimension reduction. PCA is

Table 4. Cluster centers generated by CFCM-D

GSE	Class	Cluster centers					Class distribution
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	
GSE200894	2	(0.655638, -0.204589, -0.451049)	(-0.636392, 0.364619, 0.271773)	-	-	-	2:4
GSE116312	3	(0.673968, -0.588919, -0.085049)	(0.421033, 0.381467, -0.802500)	(-0.796594, 0.464265, 0.332329)	-	-	1:5:7
GSE97332	2	(0.660730, -0.428987, -0.231743)	(-0.465838, 0.507546, -0.041708)	-	-	-	7:7
GSE159518	2	(-0.469119, -0.073515, 0.542633)	(0.685076, -0.468968, -0.216108)	-	-	-	6:10
GSE33126	2	(-0.667211, 0.482319, 0.184892)	(0.180677, 0.443360, -0.624037)	-	-	-	8:10
GSE185770	2	(0.639254, -0.187183, -0.452071)	(-0.313079, 0.709167, -0.396088)	-	-	-	8:10
GSE168466	4	(0.340406, 0.452823, -0.793229)	(0.638507, -0.679189, 0.040683)	(-0.748157, 0.543775, 0.204382)	(-0.479024, -0.282672, 0.761696)	-	7:10:10:13
GSE14520	2	(0.647228, -0.423773, -0.223455)	(-0.764190, 0.275781 -0.488409)	-	-	-	20:21
GSE50161	5	(-0.777643, 0.505724, 0.271918)	(0.646773, 0.088817, -0.735590)	(0.287328, 0.508341, -0.795669)	(0.442162, -0.804837, 0.362675)	(0.768733, -0.589436, -0.179297)	16:19:23:29:43

Table 5. GSE200894: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2
GSM6046079	0.492593	0.007407
GSM6046080	0.528644	0.008629
GSM6046081	0.010545	0.806320
GSM6046082	0.102980	0.663458
GSM6046083	0.141890	0.560613
GSM6046084	0.010161	0.812191

a projection-based approach. It searches for the linear combination of uncorrelated features with maximum variance. Here, data transformation occurs with data projection into a three-dimensional subspace with maximized variance to be followed by the data standardization process. The standardization process restricts data on the unit hypersphere \mathbb{S}^2 in \mathbb{R}^3 . Figure 1 provides a graphic of the data distribution in \mathbb{R}^3 , for the datasets GSE33126, GSE168466, GSE14520, and GSE50161 recorded in Table 1.

Table 6. GSE116312: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2	Cluster 3
GSM3227907	0.934343	0.000143	0.000214
GSM3227908	0.901138	0.002733	0.009100
GSM3227909	0.933137	0.000244	0.000332
GSM3227910	0.928254	0.000364	0.000821
GSM3227911	0.934575	0.000116	0.000184
GSM3227912	0.931579	0.000386	0.000491
GSM3227913	0.933280	0.000232	0.000318
GSM3227914	0.004585	0.391242	0.106760
GSM3227915	0.002566	0.465127	0.032307
GSM3227916	0.002037	0.478222	0.020669
GSM3227917	0.027215	0.561249	0.025586
GSM3227918	0.001371	0.500756	0.007774
GSM3227919	0.000092	0.000227	0.619966

Table 7. GSE97332: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2
GSM2561829	0.639068	0.008950
GSM2561830	0.646683	0.016250
GSM2561831	0.637882	0.009202
GSM2561832	0.426759	0.204186
GSM2561833	0.639029	0.008958
GSM2561834	0.399649	0.227705
GSM2561835	0.639655	0.008831
GSM2561836	0.016219	0.484112
GSM2561837	0.030720	0.499389
GSM2561838	0.016146	0.483854
GSM2561839	0.198176	0.302427
GSM2561840	0.016203	0.484059
GSM2561841	80.205642	0.296664
GSM2561842	0.199919	0.301081

CFCM-D, FCM and HFCM were run with different experimental parameters. The CFCM-D algorithm uses two experimental parameters, m and ϑ to estimate robust partitioning of the data; $m \in \{1.1, 2\}$ and $\vartheta \in \{0.1, 0.2, 0.4, 0.5\}$. Table 2 records the data-specific experimental parameters used in the CFCM-D algorithm. $c \in \{2, 3, 4, 5\}$. In FCM, the parameter m was set as 2 for all

Table 8. GSE159518: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2
GSM4831674	0.218220	0.532681
GSM4831675	0.024816	0.640776
GSM4831676	0.020503	0.689870
GSM4831677	0.077927	0.671487
GSM4831678	0.170299	0.348920
GSM4831679	0.105673	0.446914
GSM4831680	0.154222	0.597753
GSM4831681	0.079575	0.494020
GSM4831682	0.025698	0.636537
GSM4831683	0.114653	0.636929
GSM4831684	0.422081	0.077919
GSM4831685	0.620363	0.005456
GSM4831686	0.617509	0.005391
GSM4831687	0.640102	0.007299
GSM4831688	0.602371	0.00559
GSM4831689	0.654931	0.012877

Table 9. GSE33126: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2
GSM820516	0.852239	0.026516
GSM820517	0.108736	0.422271
GSM820518	0.830751	0.062511
GSM820519	0.028156	0.701198
GSM820520	0.816428	0.018163
GSM820521	0.069520	0.772046
GSM820522	0.649446	0.059546
GSM820523	0.729943	0.035183
GSM820524	0.516097	0.109775
GSM820525	0.156691	0.343309
GSM820526	0.848642	0.020699
GSM820527	0.450896	0.440407
GSM820528	0.831727	0.061347
GSM820529	0.105096	0.750774
GSM820530	0.728450	0.169725
GSM820531	0.028992	0.753888
GSM820532	0.069190	0.518114
GSM820533	0.244465	0.633878

the datasets, while in HFCM, m was set as 2 for all the datasets except for GSE33126, $m = 3$. CFCM-D, FCM and HFCM algorithms were run 20 times for a set of experimental parameters and the best solution was recorded. The termination condition is $\max_{\text{iteration}} = 150$ or $\sum_{i=1}^n \sum_{k=1}^c |u_{ik}^{t+1} - u_{ik}^t| < 10^{-5}$, t is an iteration number.

Table 10. GSE185770: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2
GSM5623408	0.321058	0.020411
GSM5623409	0.516745	0.008416
GSM5623410	0.052589	0.770102
GSM5623411	0.022454	0.796871
GSM5623412	0.666854	0.017659
GSM5623413	0.036386	0.785801
GSM5623414	0.548553	0.007533
GSM5623415	0.064932	0.376601
GSM5623416	0.652678	0.100642
GSM5623417	0.060129	0.039871
GSM5623418	0.193080	0.028631
GSM5623419	0.049218	0.773476
GSM5623420	0.009570	0.764786
GSM5623421	0.098195	0.031250
GSM5623422	0.019493	0.674709
GSM5623423	0.015484	0.704317
GSM5623424	0.053272	0.460623
GSM5623425	0.051468	0.771228

Table 4 records the cluster centers predicted by the CFCM-D algorithm. CFCM-D generates noise-resistant non-coincident cluster centers:

$$v_k = \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_i \left\{ \sum_{j=1}^p \left\{ \sum_{i=1}^n w_i u_{ik}^m \tilde{x}_{ij} \right\}^2 \right\}^{-\frac{1}{2}}, 1 \leq k \leq c,$$

w_i in v_k factors in $\min_{r=1,2,\dots,n} \kappa_r$ with each feature vector \tilde{x}_i . $\min_{r=1,2,\dots,n} \kappa_r$ ascertains the number of feature vectors in the immediate neighbourhood of \tilde{x}_i . $\kappa_i = \{1 - \sum_{j=1}^p \tilde{x}_{ij} \varrho_j\}$; wherein ϱ is a central representative of \tilde{x}_i . w_i of \tilde{x}_i corresponds to $|\tilde{x}_l|_{l \neq i}^n$, $\tilde{x}_l \in \{\tilde{x}_i\}_{i=1}^n$ closer than $\min_{r=1,2,\dots,n} \kappa_r$; for an outlier, $\tilde{x}_N, w_N \sim \frac{1}{\infty}$. Thus, the cluster centers v_k account for the data distribution in space. $\{v_k\}_{k=1}^c$ is noise-resistant.

Tables 5, 6, 7, 8, 9, 10, 11, 12, and 13 record the membership assignments generated by CFCM-D. In CFCM-D, $\sum_{k=1}^c u_{ik} \geq \vartheta$. ϑ controls the minimum of ψ_i ; for the noisiest point, $\psi_i = 1 - \frac{\{1-\vartheta\}\kappa_i}{\max_{r=1,2,\dots,n} \kappa_r} = 1 - \frac{\{1-\vartheta\}}{\max_{r=1,2,\dots,n} \frac{\kappa_r}{\kappa_r}} = 1 - \{1 - \vartheta\} = \vartheta$. $0 < \vartheta < 1$. This ensures $\sum_{k=1}^c u_{ik} \geq \vartheta$. Further, $\sum_{k=1}^c u_{ik} \neq 1$. The range of the membership assignments generated with CFCM-D is provided in Table 3.

Table 11. GSE14520: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2	Accession	Cluster 1	Cluster 2
GSM362947	0.407587	0.004501	GSM363430	0.392849	0.008138
GSM362948	0.431587	0.006018	GSM363431	0.000807	0.633349
GSM362950	0.456204	0.021912	GSM363432	0.440741	0.008448
GSM362951	0.440423	0.008334	GSM363433	0.000563	0.640233
GSM362952	0.000846	0.642884	GSM363434	0.457547	0.027721
GSM362953	0.335381	0.097077	GSM363435	0.000566	0.640423
GSM362954	0.007275	0.598836	GSM363436	0.330430	0.104547
GSM362955	0.446538	0.011086	GSM363437	0.003857	0.639422
GSM362955	0.000908	0.631974	GSM363438	0.452611	0.015936
GSM362956	0.426651	0.005232	GSM363439	0.000566	0.638792
GSM362957	0.001808	0.623692	GSM363440	0.344613	0.082874
GSM363420	0.400207	0.005450	GSM363441	0.000673	0.642170
GSM363421	0.000627	0.636651	GSM363442	0.388985	0.011015
GSM363422	0.422014	0.004734	GSM363443	0.001602	0.642846
GSM363423	0.001412	0.626801	GSM363444	0.427308	0.005321
GSM363424	0.373261	0.035270	GSM363445	0.006347	0.635229
GSM363425	0.000727	0.634626	GSM363446	0.371743	0.037957
GSM363426	0.405982	0.004621	GSM363447	0.000670	0.635685
GSM363427	0.003414	0.640161	GSM363448	0.153812	0.351155
GSM363428	0.450573	0.058247	GSM363449	0.001298	0.643091
GSM363429	0.000723	0.642472			

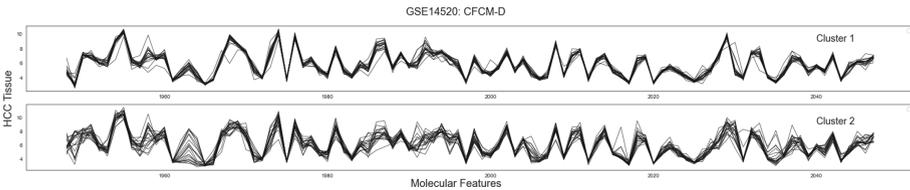


Fig. 2. GSE14520: The molecular signature of HCC tissue as clustered with CFCM-D (100 features)

Table 12. GSE168466: Membership assignments with CFCM-D

Accession	Cluster 1	Cluster 2	Cluster 3	Cluster 4
GSM5140954	0.528604	0.000519	0.000327	0.000143
GSM5140955	0.035285	0.454299	0.004011	0.006405
GSM5140956	0.525514	0.000851	0.000453	0.000212
GSM5140957	0.071955	0.413379	0.006150	0.008673
GSM5140958	0.271434	0.211214	0.010098	0.010521
GSM5140959	0.001727	0.503913	0.000630	0.001954
GSM5140960	0.533040	0.000301	0.000245	0.000097
GSM5140961	0.001693	0.505521	0.000706	0.002413
GSM5140962	0.538575	0.000288	0.000329	0.000112
GSM5140963	0.006874	0.467429	0.006657	0.048304
GSM5140964	0.531385	0.000355	0.000262	0.000107
GSM5140965	0.012704	0.482143	0.002050	0.003883
GSM5140966	0.543396	0.000509	0.000808	0.000239
GSM5140967	0.001701	0.505653	0.000719	0.002480
GSM5140968	0.537708	0.000275	0.000297	0.000104
GSM5140969	0.002719	0.505612	0.001645	0.007582
GSM5140970	0.535198	0.000268	0.000248	0.000093
GSM5140971	0.002985	0.498429	0.000784	0.001958
GSM5140972	0.540449	0.000340	0.000438	0.000142
GSM5140973	0.003879	0.499838	0.002799	0.015121
GSM5140974	0.002214	0.000649	0.604799	0.004259
GSM5140975	0.010208	0.204475	0.019105	0.314818
GSM5140976	0.001213	0.000477	0.608135	0.004445
GSM5140977	0.000912	0.004538	0.005372	0.569616
GSM5140978	0.004259	0.003899	0.462355	0.143066
GSM5140979	0.002459	0.003931	0.075334	0.524567
GSM5140980	0.004679	0.005169	0.318411	0.283448
GSM5140981	0.256931	0.012635	0.295495	0.021165
GSM5140982	0.004679	0.005169	0.318512	0.283349
GSM5140983	0.040925	0.005002	0.539643	0.014825
GSM5140984	0.002857	0.000773	0.602824	0.004651
GSM5140985	0.000497	0.001593	0.004958	0.584349
GSM5140986	0.001175	0.000506	0.607774	0.005301
GSM5140987	0.000811	0.003829	0.005048	0.572048
GSM5140988	0.001502	0.000511	0.607254	0.003977
GSM5140989	0.003021	0.002269	0.552645	0.056922
GSM5140990	0.004464	0.004271	0.432844	0.171617
GSM5140991	0.001070	0.002250	0.019993	0.577683
GSM5140992	0.002595	0.000723	0.603606	0.004489
GSM5140993	0.001479	0.008953	0.007103	0.557953

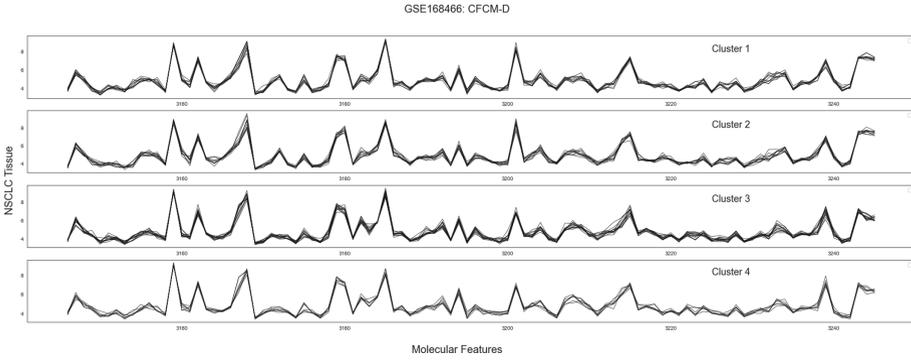


Fig. 3. GSE168466: The molecular signature of NSCLC tissues as clustered with CFCM-D (100 features)

Table 13. GSE50161: Membership assignments with CFCM-D (20 Instances)

Accession	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
GSM1214834	0.244487	0.000639	0.000986	0.001208	0.000653
GSM1214835	0.391442	0.002280	0.006446	0.001180	0.001113
GSM1214836	0.309934	0.000122	0.000238	0.000128	0.000090
GSM1214837	0.365917	0.000615	0.001509	0.000410	0.000350
GSM1214838	0.357379	0.000422	0.000996	0.000302	0.000250
GSM1214839	0.284408	0.000176	0.000315	0.000230	0.000147
GSM1214840	0.287482	0.000163	0.000294	0.000207	0.000134
GSM1214841	0.329672	0.000158	0.000333	0.000141	0.000107
GSM1214842	0.260120	0.000378	0.000617	0.000615	0.000356
GSM1214843	0.314186	0.000123	0.000246	0.000125	0.000089
GSM1214844	0.168861	0.004007	0.004811	0.016285	0.006037
GSM1214845	0.366020	0.000618	0.001517	0.000412	0.000351
GSM1214846	0.299663	0.000130	0.000245	0.000148	0.000101
GSM1214847	0.013722	0.030329	0.547154	0.002143	0.003946
GSM1214848	0.279300	0.000204	0.000356	0.000278	0.000174
GSM1214849	0.346265	0.000269	0.000606	0.000211	0.000168
GSM1214850	0.323218	0.000137	0.000283	0.000129	0.000096
GSM1214851	0.201563	0.002163	0.002869	0.006325	0.002769
GSM1214852	0.350379	0.000316	0.000724	0.000239	0.000194
GSM1214853	0.273459	0.000243	0.000417	0.000350	0.000215

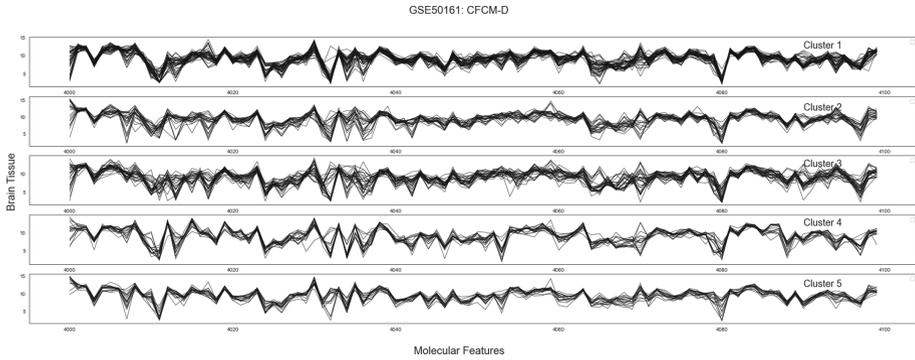


Fig. 4. GSE50161: The molecular signature of brain tissues as clustered with CFCM-D (100 features)

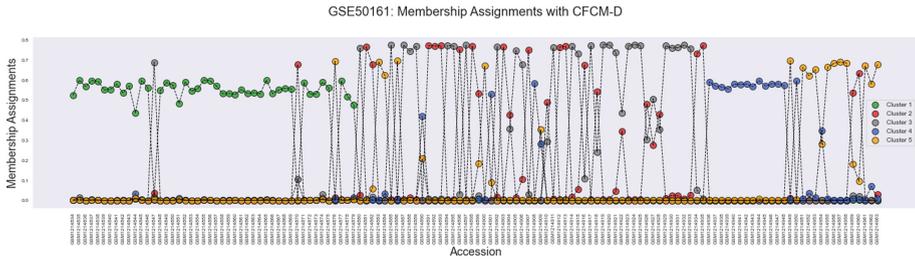


Fig. 5. GSE50161: Membership assignments with CFCM-D

Table 14. Performance Measure: Precision and Recall

Measure	Dataset	FCM	HFCM	CFCM-D
Precision	GSE200894	0.8333	0.8333	0.8333
	GSE116312	0.7692	0.6932	0.8462
	GSE97332	1.0000	0.6429	1.0000
	GSE159518	1.0000	0.2500	1.0000
	GSE33126	0.5000	0.8333	0.8888
	GSE185770	0.6666	0.6111	0.6111
	GSE168466	0.2500	0.5250	0.9250
	GSE14520	0.9756	0.4390	0.9756
Recall	GSE50161	0.6077	0.4154	0.7462
	GSE200894	0.8333	0.8333	0.8333
	GSE116312	0.7692	0.6932	0.8462
	GSE97332	0.5714	0.6429	1.0000
	GSE159518	0.8333	0.2500	1.0000
	GSE33126	0.5000	0.8333	0.8888
	GSE185770	0.6666	0.6111	0.6111
	GSE168466	0.2500	0.5250	0.9250
GSE14520	0.9756	0.4390	0.9756	
GSE50161	0.6077	0.4154	0.7462	

Table 15. Performance Measure: Internal Measure

Measure	Dataset	FCM	HFCM	CFCM-D
SC	GSE200894	0.2674	0.0462	0.4744
	GSE116312	0.4495	0.3341	0.6789
	GSE97332	0.4274	0.4926	0.5140
	GSE159518	0.4976	0.2277	0.4993
	GSE33126	0.3360	0.1905	0.4085
	GSE185770	0.3240	0.2026	0.4902
	GSE168466	0.6706	0.2444	0.5912
	GSE14520	0.4883	0.2474	0.6782
	GSE50161	0.5085	0.0870	0.6626
DB	GSE200894	1.0732	1.5909	0.5998
	GSE116312	0.9860	1.0359	0.2750
	GSE97332	0.8868	1.5145	0.8135
	GSE159518	0.7880	1.3339	0.7561
	GSE33126	1.2066	1.7784	0.9888
	GSE185770	1.1791	1.4738	0.8447
	GSE168466	0.4525	1.3421	0.4905
	GSE14520	0.8184	1.5088	0.4885
	GSE50161	0.7173	2.1997	0.4444
CH	GSE200894	3.2233	1.3574	6.6762
	GSE116312	8.8521	6.1192	51.3275
	GSE97332	6.5680	6.6698	12.9559
	GSE159518	13.8546	5.5768	15.0679
	GSE33126	8.4218	4.5757	13.6756
	GSE185770	8.6890	4.3638	19.0143
	GSE168466	95.7104	16.6969	86.9274
	GSE14520	38.5006	6.4106	106.3362
	GSE50161	183.8097	18.0515	632.7249

The experiments with FCM, HFCM and CFCM-D are described in Tables 14 and 15. The data recorded in Table 1 is associated with the class information (c and a class distribution); this provides a chance to assess the robustness of CFCM-D by external measures. Here, precision and recall are used. Table 14 records the precision and recall for FCM, HFCM and CFCM-D. In comparison to FCM and HFCM, CFCM-D obtained the optimal partitioning for the datasets GSE200894, GSE116312, GSE97332, GSE159518, GSE33126, GSE168466, GSE14520, and GSE50161 according to the precision and recall measures. CFCM-D partitions GSE97332 and GSE158519 with a precision of 1.0000. The performance of CFCM-D was inferior for GSE185770. CFCM-D is

partitioned GSE185770 with a precision of 0.6111 with seven misclassification; 8.3258 % decrease in precision is noted when CFCM-D is used over the standard FCM algorithm. The performance of these algorithms are further assessed with three internal measures: SC, DB and CH. Table 15 records the SC, DB and CH measures for FCM, HFCM and CFCM-D. It is observed that CFCM-D has the optimal SC, DB and CH scores for all the datasets except for GSE168466.

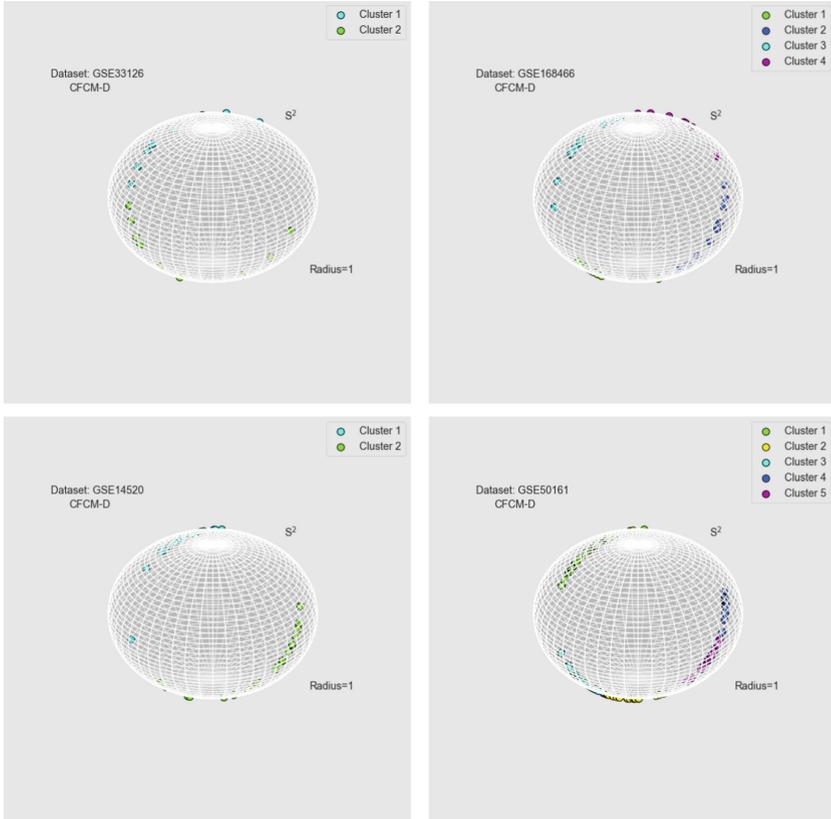


Fig. 6. Data partitioning with CFCM-D

The data clustering procedure with CFCM-D is summarized in Figs. 2, 3, 4, 5 and 6. Figure 6 provides a graphical illustration of the data partitioning with CFCM-D for the datasets GSE33126, GSE168466, GSE14520, and GSE50161. CFCM-D partitions data on S^2 in \mathbb{R}^3 . S^2 is a hypersphere with unit radius, $c \in \{2, 3, 4, 5\}$. The molecular signature of disease subgroups differs more in the direction than in magnitude of gene expression, which is captured in Figs. 2, 3 and 4. The graphical illustrations in Figs. 2, 3 and 4 record the molecular signature of the data (GSE14520, GSE168466, GSE50161) clustered by CFCM-D. These

molecular signatures are recorded over 100 random features; GSE14520: 1948–2048, GSE168466: 3145–3245, GSE50161: 4000–4100. Figure 5 provides a graphical illustration of the membership assignments with CFCM-D for the dataset GSE50161, a feature vector is assigned to a cluster with the highest membership.

7 Conclusion

This paper presented a noise-resistant fuzzy-based algorithm (CFCM-D) to partition cancer data. The article discusses a data standardization process that generates directional data. The proposed CFCM-D algorithm groups the standardized data in order to discover clusters insensitive to noise. The proposed approach is effective in partitioning directional data; the obtained results on the nine expression datasets can be considered satisfactory. The experiments show that CFCM-D outperforms the HFCM algorithm, which is in common use for partitioning directional data. At the same time, the performance of CFCM-D was found comparable with the conventional FCM.

Future research can focus on the application of the proposed mechanism to other real-world classification problems beyond bio-molecular data. Directional data is encountered in the domain of document clustering wherein document vectors are normalized to remove a bias arising from a document's length. The proposed approach can be used for grouping documents with noise. Documents with noise are automated texts generated by speech or character recognition. The proposed approach can further be used for sentiment analysis of opinions expressed on Twitter, product reviews, movie reviews, YouTube comments, or travel blogs.

Acknowledgment. Oleksii K. Tyshchenko acknowledges the Helmholtz Information and Data Science Academy (HIDA) for providing financial support enabling a short-term research stay at Helmholtz Centre for Infection Research (Braunschweig, Germany) for fuzzy clustering analysis of biological data.

References

1. Babichev, S., Skvor, J., Fiser, J., Lytvynenko, V.: Technology of gene expression profiles filtering based on wavelet analysis. *Int. J. Intell. Syst. Appl. (IJISA)* **10**(4), 1–7 (2018). <https://doi.org/10.5815/ijisa.2018.04.01>
2. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S., Ridgeway, G.: Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.* **6**(9), 1345–1382 (2005)
3. Banerjee, A., Ghosh, J.: Frequency sensitive competitive learning for clustering on high-dimensional hyperspheres. In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN 2002 (Cat. No. 02CH37290)*, vol. 2, pp. 1590–1595. IEEE (2002). <https://doi.org/10.1109/IJCNN.2002.1007755>
4. Banerjee, A., Ghosh, J.: Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres. *IEEE Trans. Neural Netw.* **15**(3), 702–719 (2004). <https://doi.org/10.1109/TNN.2004.824416>

5. Benjamin, J.B., Hussain, I., Yang, M.S.: Possibilistic c-means clustering on directional data. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–6. IEEE (2019). <https://doi.org/10.1109/CISP-BMEI48845.2019.8965703>
6. Bezdek, J.C.: Fuzzy-mathematics in pattern classification. Cornell University (1973)
7. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. *Commun. Stat.-Theory Methods* **3**(1), 1–27 (1974). <https://doi.org/10.1080/03610927408827101>
8. Callari, M., et al.: Comparison of microarray platforms for measuring differential microrna expression in paired normal/cancer colon tissues (2012). <https://doi.org/10.1371/journal.pone.0045105>
9. Chintalapudi, K.K., Kam, M.: A noise-resistant fuzzy c means algorithm for clustering. In: 1998 IEEE international conference on fuzzy systems proceedings. In: IEEE World Congress on Computational Intelligence (Cat. No. 98CH36228), vol. 2, pp. 1458–1463. IEEE (1998). <https://doi.org/10.1109/FUZZY.1998.686334>
10. Chintalapudi, K.K., Kam, M.: The credibilistic fuzzy c means clustering algorithm. In: SMC 1998 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218), vol. 2, pp. 2034–2039. IEEE (1998). <https://doi.org/10.1109/ICSMC.1998.728197>
11. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, 224–227 (1979). <https://doi.org/10.1109/TPAMI.1979.4766909>
12. Dortet-Bernadet, J.L., Wicker, N.: Model-based clustering on the unit sphere with an illustration using gene expression profiles. *Biostatistics* **9**(1), 66–80 (2008). <https://doi.org/10.1093/biostatistics/kxm012>
13. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters (1973). <https://doi.org/10.1080/01969727308546046>
14. Gao, X., Xie, W.: Advances in theory and applications of fuzzy clustering. *Chin. Sci. Bull.* **45**(11), 961–970 (2000). <https://doi.org/10.1007/BF02884971>
15. Griesinger, A.M.: Characterization of distinct immunophenotypes across pediatric brain tumor types. *J. Immunol.* **191**(9), 4880–4888 (2013). <https://doi.org/10.4049/jimmunol.1301966>
16. Hornik, K., Feinerer, I., Kober, M., Buchta, C.: Spherical k-means clustering. *J. Stat. Softw.* **50**, 1–22 (2012)
17. Kaur, P.: Intuitionistic fuzzy sets based credibilistic fuzzy C-means clustering for medical image segmentation. *Int. J. Inf. Technol.* **9**(4), 345–351 (2017). <https://doi.org/10.1007/s41870-017-0039-2>
18. Kaur, P., Soni, A., Gosain, A.: Robust kernelized approach to clustering by incorporating new distance measure. *Eng. Appl. Artif. Intell.* **26**(2), 833–847 (2013). <https://doi.org/10.1016/j.engappai.2012.07.002>
19. Kesemen, O., Tezel, Ö., Özkul, E.: Fuzzy c-means clustering algorithm for directional data (fcm4dd). *Expert Syst. Appl.* **58**, 76–82 (2016). <https://doi.org/10.1016/j.eswa.2016.03.034>
20. Lashkmaiah, K., Murali Krishna, S., Eswara Reddy, B.: An optimized k-means with density and distance-based clustering algorithm for multidimensional spatial databases. *Int. J. Comput. Netw. Inf. Secur. (IJCNIS)* **13**(6), 70–82 (2021). <https://doi.org/10.5815/ijcnis.2021.06.06>
21. Maitra, R., Ramler, I.P.: A k-mean-directions algorithm for fast clustering of data on the sphere. *J. Comput. Graph. Stat.* **19**(2), 377–396 (2010). <https://doi.org/10.1198/jcgs.2009.08155>

22. Mendes, M., Sacks, L.: Evaluating fuzzy clustering for relevance-based information access. In: The 12th IEEE International Conference on Fuzzy Systems, 2003, FUZZ 2003, vol. 1, pp. 648–653. IEEE (2003). <https://doi.org/10.1109/FUZZ.2003.1209440>
23. Nguyen, T., Khosravi, A., Creighton, D., Nahavandi, S.: Hierarchical gene selection and genetic fuzzy system for cancer microarray data classification. *PloS one* **10**(3), e0120364 (2015). <https://doi.org/10.1371/journal.pone.0120364>
24. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. *Science* **344**(6191), 1492–1496 (2014). <https://doi.org/10.1126/science.1242072>
25. Roessler, S., et al.: A unique metastasis gene signature enables prediction of tumor relapse in early-stage hepatocellular carcinoma patients. *Cancer Res.* **70**(24), 10202–10212 (2010). <https://doi.org/10.1158/0008-5472.CAN-10-2607>
26. Starczewski, A., Krzyżak, A.: Performance evaluation of the silhouette index. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2015. LNCS (LNAI), vol. 9120, pp. 49–58. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19369-4_5
27. Yan, Y., Chen, L.: Hyperspherical possibilistic fuzzy c-means for high-dimensional data clustering. In: 2009 7th International Conference on Information, Communications and Signal Processing (ICICSP), pp. 1–5. IEEE (2009). <https://doi.org/10.5555/1818318.1818479>
28. Yang, M.S., Chang-Chien, S.J., Hung, W.L.: An unsupervised clustering algorithm for data on the unit hypersphere. *Appl. Soft Comput.* **42**, 290–313 (2016). <https://doi.org/10.1016/j.asoc.2015.12.037>
29. Yang, M.S., Pan, J.A.: On fuzzy clustering of directional data. *Fuzzy Sets Syst.* **91**(3), 319–326 (1997). [https://doi.org/10.1016/S0165-0114\(96\)00157-1](https://doi.org/10.1016/S0165-0114(96)00157-1)
30. Yang, M.S., Wu, K.L.: A similarity-based robust clustering method. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(4), 434–448 (2004). <https://doi.org/10.1109/TPAMI.2004.1265860>
31. Yu, X., Yu, G., Wang, J.: Clustering cancer gene expression data by projective clustering ensemble. *PloS one* **12**(2), e0171429 (2017). <https://doi.org/10.1371/journal.pone.0171429>
32. Zeng, Y., Xu, Z., He, Y., Rao, Y.: Fuzzy entropy clustering by searching local border points for the analysis of gene expression data. *Knowl.-Based Syst.* **190**, 105309 (2020). <https://doi.org/10.1016/j.knosys.2019.105309>



Visual Analytics-Based Method for Sentiment Analysis of COVID-19 Ukrainian Tweets

Oleksii Kovalchuk¹, Vitalii Slobodzian¹, Olena Sobko¹,
Maryna Molchanova¹, Olexander Mazurets¹, Oleksander Barmak¹,
Iurii Krak^{2,3}✉, and Nataliia Savina⁴

¹ Khmelnytskyi National University, Khmelnytskyi, Ukraine

² Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

`krak@univ.kiev.ua, yuri.krak@gmail.com`

³ Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv, Ukraine

⁴ National University of Water and Environmental Engineering, Rivne, Ukraine

`n.b.savina@nuwm.edu.ua`

Abstract. The coronavirus pandemic is one of the leading communication topics for users on social networks. It causes different emotions in people: fear, sadness, anger, joy, and elation. Detecting sentiment about the pandemic is an acute challenge because it helps track people's attitudes about the pandemic itself and the messages and decisions of local authorities aimed at combating the coronavirus. To address the issue, namely natural language processing, messages are processed using the TextRank vectorization method and the SVM-based two-level classification model. The first stage is the detection of tweets that are directly related to the coronavirus. The second stage means detecting the sentiment of the dataset obtained in the first stage. The classifier's effectiveness was tested using the following metrics: precision, recall, F1-norm, and confusion matrix, and averaged about 90%. Thus, the automated detection of the sentiment of Twitter messages about the coronavirus pandemic was obtained. The approach described in the paper will allow assessing public opinion on pandemic control measures applied by the country's governments.

Keywords: Twitter · COVID-19 · Sentiment · Text data
vectorization · TextRank · SVM

1 Introduction

Social networks, which have emerged due to the rapid development of information technology, significantly impact people's lives. In social networks, people

are used to sharing personal information and expressing their opinions and attitudes on various phenomena and events. Recently, it is also being developed alternative communication technologies that involve people with special needs have [17], which also expanded the audience and scope of communication [21]. Such availability and ease of collecting and processing information from social networks make them a popular platform for various scientific studies.

One of the phenomena that significantly changed the social life of people around the world was the COVID-19 pandemic. To prevent the spread of the disease, the governments of different countries introduce strict quarantine conditions, which primarily involve the concept of self-isolation. The main slogan of the fight against coronavirus disease is “stay at home.” In addition to the physical effects on people, the coronavirus also affects mental health. Studies indicate that coronavirus disease has also led to an increase in loneliness [13]. Looking for opportunities not to be alone, more people began to use social networks to replace real communication, which becomes limited while quarantining. In addition, the phenomenon of the COVID-19 pandemic has become highly discussed in the network.

Users of social networks express their views on the consequences of the pandemic, measures to prevent the spread of the disease, and ways to fight. Thanks to the collection of relevant data, the task of identifying people’s attitudes towards the COVID-19 pandemic is becoming actual. Such information can be helpful in various sociological surveys and for use by local authorities.

Investigating users’ emotions is crucial because it can investigate how people feel during a pandemic. These data may influence government decisions, such as increased support for education, medicine, or media coverage of the coronavirus [19].

Defining public opinion and understanding people’s attitudes towards the pandemic are highly topical. It is crucial to understand and assess how satisfied society is with the actions of central and health authorities to combat the pandemic and overcome the effects of the disease [3]. Today, the best source of people’s opinions is social networks, particularly online micro-blogs. Accordingly, the automation of semantic analysis based on micro-blogging on people’s attitudes to the coronavirus pandemic seems an urgent task. The main contributions of this paper are summarized as follows:

- The paper investigates approaches to sentiment analysis and proposes a method of automated determination of emotional sentiments on texts in social networks, which can perform multi-level classification of short text messages.
- The peculiarity of the proposed method is its focus on textual data presented in the Ukrainian language, which is characterized by different from other languages’ stylistic and semantic features.
- Two primary advantages of the proposed method over analogs are the categorization of emotional attitudes by semantic features of the Ukrainian literary language and psycho-emotional features of the Ukrainian jargon language, and the human-based approach that allows robust interpretation of classification results.

- In the more extensive use, our method provides the ability to target collective and individual behavioral anomalies for their timely blocking, which has significant social and political weight.
- Studies conducted to classify short text messages in Ukrainian from Twitter by semantic and emotional characteristics of the content revealed an average classification of 90%, which is competitive with state-of-the-art approaches.

The rest of the paper is structured in the following way. Section 2 reviews recent publications correlated with the challenges of sentiment analysis of short texts. Section 3 addresses the issue of defining public opinion and understanding people’s attitudes towards the COVID-19 pandemic based on micro-blogging in Ukrainian and formalizes the problem statement. Section 4 describes the proposed visual analytics-based method for sentiment analysis of COVID-19 Ukrainian tweets and formalizes the ways of its assessment. Section 5 includes the description of the computational experiments and presents the results of categorizing short texts in Ukrainian into positive and negative semantics. *Section “Conclusions”* summarizes the research results and indicates the direction for future work.

2 Literature Review

Next, a review of recent publications correlated with the described challenge was conducted. In [5], the authors analyze two types of tweets from different time intervals. Datasets were created by searching for messages that contained the hashtags #corona, #covid19, #coronavirus, #COVID-19. In their work, the authors compare several models for determining the sentiments of tweets (positive, neutral, negative). This paper proposes the model based on fuzzy logic implemented using a Support Vector Machine (SVM) with an accuracy of 79%.

In work [12], it was investigated the attitude of social network users to the statements and actions of local authorities at the beginning of the pandemic and during the increase in mortality. Despite the outbreaks of deaths, users responded positively to reports and actions by the authorities. The authors used SVM with the Naive Bayes classifier in their study.

In study [23], the authors tracked the change in attitudes of Twitter users regarding the number of new cases of infection in the countries most affected by the coronavirus. The authors use three sets of messages related to work at home, online learning, and just tweets related to COVID-19. To determine sentiments, two models are compared - LSTM and ANN [26]. The LSTM was 84.5% accurate, while the ANN achieved an accuracy of 76%.

The authors of [28] compare five models of supervised machine learning, such as RF, XGBoost, SVC, ETC, and DT, and 1 model of deep learning LSTM to determine the sentiment of coronavirus messages. The ETC model using the combination of TF-IDF and BoW gave the best result. The LSTM model showed the worst results.

In [11], the authors presented their approach to classifying Twitter messages related to COVID-19 into positive, negative, and neutral. The mood classification was performed using RNN and SVM, presented its model called Hybrid Heterogeneous Support Vector Machine [20]. The RNN model gave the best polarization of the results, while SVM identified most tweets as neutral.

The RNN model was used to classify the mood of coronavirus reports in study [27]. The authors of the study compare the results by utilizing the TextBlob model. The results using the RNN model appeared better than TextBlob.

In research [29], the authors used the Latent Dirichlet Allocation model to determine the attitudes of Twitter users towards COVID-19. The study found that users were filled with feelings of hope, sadness, and fear and did not give up hope for improvement.

The authors of article [24] investigated the sentiments of Twitter users regarding the actions of the Indonesian authorities to combat the coronavirus. The study used the SVM model, which gave high-performance models, namely precision, recall, and F1-measure, with averages of 82%.

The task of classifying the emotions of users of social networks that caused them to report WHO about the coronavirus was highlighted by the authors of article [9]. A set of messages was collected containing WHO and COVID-19, 4,000 tweets with positive emotions and 4,000 with negative ones. The classifier model proposed by the authors combined SVM, N-Gram, and Area Under ROC Curve and achieved high efficiency.

For detecting emotions, the authors [30] use the TextRank method to extract keywords. In this paper, the authors conduct an empirical study of TextRank to find the optimal settings for keyword extraction.

There are also other approaches to visualizing and interpreting machine learning results. For instance, the authors in [25] suggested a manually crafted neural network with convolutional kernels of $3 \times 3 \times 3$ to resolve the task of segmenting abdominal organs in computed tomography images.

The authors of the article were able to classify the feelings of anger, sadness, fear, and joy about the messages about the coronavirus on Twitter [7]. Data analysis was performed using the BERT model, which was compared with three other models, such as LR, SVM, and LSTM. The BERT model gave 89% accuracy, the best result compared to other models. As it is seen from the analysis above, the SVM method can be considered a reliable approach for constructing a linear hyperplane classification and prediction [15, 18]. It is considered an effective technique for classifying substantial amounts of data. Also, the training of the linear SVM classifier is faster than nonlinear [6].

Previous studies focused on the visual analytics approach based on the MDS method to display vectorization results [14]. This method is a means of evaluating vectorization to obtain a linear classification model. The TF-IDF, variance estimation, TextRank, and YAKE algorithms were researched for keyword searching in messages [16].

TextRank was selected for keyword searching in messages because it provides the best results [31]. TextRank is a text analysis algorithm that is based on the

PageRank algorithm. This algorithm is widely used to search keywords in texts. It is based on a graph model [22], in which the relationship between words is represented as the edges of this graph.

Based on the results of the analysis of related works, it was concluded that the existing approaches and means of textual analysis do not allow for effective semantic analysis of tweets written in Ukrainian. Therefore, the study aims to develop a method of automated detection of positive or negative attitudes of social network users based on short text messages in Ukrainian on the COVID-19 pandemic.

3 Problem Statement

The goal of this study is to identify sentiment on short messages about the coronavirus among Ukrainian-speaking users on Twitter. Analysis of the textual content of social networks imposes several limitations using standard approaches to text mining, which are defined as follows:

- The Ukrainian language written on the Internet is not literary. It contains many words borrowed from other languages and the so-called surzhyk. Such a plain version of the Ukrainian language can create obstacles to correctly classifying moods.
- The size of tweets is limited to 280 characters, making it difficult to use standard means of detecting meaning in texts.
- There are few freely accessible and statistically verified corpora for the Ukrainian language with authentic messages.
- Tweets can contain grammatical mistakes, which also affect the correctness of the classification.
- There are few ready-made solutions to this problem for Ukrainian-language content.

The mentioned above challenges form the issues that must be resolved within the study. To do so, it is considered adequate to use the TextRank method to extract keywords from messages and the SVM classifier to analyze data to classify messages of Twitter users into negative and positive, related to the topic of coronavirus. The advantages of the proposed method of detecting the sentiment of messages are that it solves the problem of classifying the Ukrainian-language content of the social network Twitter, which is characterized by flexibility, use of dialect, surzhyk, and twisting words.

4 Materials and Methods

A robust feature extraction approach and an efficient classification model are crucial to achieving the study's primary goal. Immediate attention was paid to constructing such a model that would allow classifying with satisfactory quality according to the linear law. A general approach to the construction of a vector of features for the classification of messages by classes: "coronavirus" - "non-coronavirus" and a vector of features for the classification of messages from the class "coronavirus" into negative and positive messages.

4.1 Preparation of Experimental Data

The dataset was created to analyze and construct classifiers. This dataset contains Ukrainian-language messages from Twitter. The dataset was created based on the two following datasets.

- TBCOV: Two Billion Multilingual COVID-19 Tweets with Sentiment, Entity, Geo, and Gender Labels is TBCOV Dataset that contains 2,014,792,896 multilingual tweets related to the COVID-19 pandemic by using 800 multilingual keywords. Tweets were collected from February 1, 2020, to March 31, 2021. Data are presented in 67 international languages, including Ukrainian. Also, tweets are labeled by geolocation and sentiment (negative, neutral, positive) [10].
- ukr-twi-corpus - the corpus of Ukrainian-language tweets with 185,499 messages [4]. The corpus is created by uploading tweets using the API Twitter Scraper [8].

Tweets from both corpora were selected to create the targeted dataset. The authors consider the created dataset sufficient for training and approbation of the classifier results.

4.2 Construction of Vector of Features (Model) for Linear Classification by Visual Analytics

The objective of this study is to detect attitudes about the coronavirus among Ukrainian speaking users. The Ukrainian language used on the Internet is not literary. It contains many words borrowed from other languages and the so-called Surzhyk. It can interfere with the correct classification of moods in messages. The authors propose a method for the models' construction (text vectorization) for the following linear classification based on visual analytics results. The main steps of the method will be described below for the following classification models: "coronavirus" - "non-coronavirus" and "coronavirus-positive emotion" - "coronavirus-negative emotion".

The training set for constructing the vector of features and training the SVM consists of four sets: $CT = \{tweet_1, tweet_2, \dots\}$ and $NCT = \{tweet_1, tweet_2, \dots\}$ sets contain tweets related to the "coronavirus" and "non-coronavirus" classes for the first classifier - $CN = \{tweet_1, tweet_2, \dots\}$ and $CP = \{tweet_1, tweet_2, \dots\}$ sets contain tweets related to the "coronavirus-positive emotion" and "coronavirus-negative emotion" classes for the second classifier.

The process consists of 6 main steps (Fig. 1) and form an intermediate stage between data preparation and further quality assessment for the prepared model and can be described in detail as follows:

- **Lemmatization of tweets.** Texts of tweets of CT and NCT , CN and CP sets are converted to a normalized form. The VESUM database was used within this stage [1]. The words that were not found in the database remained in their original form.

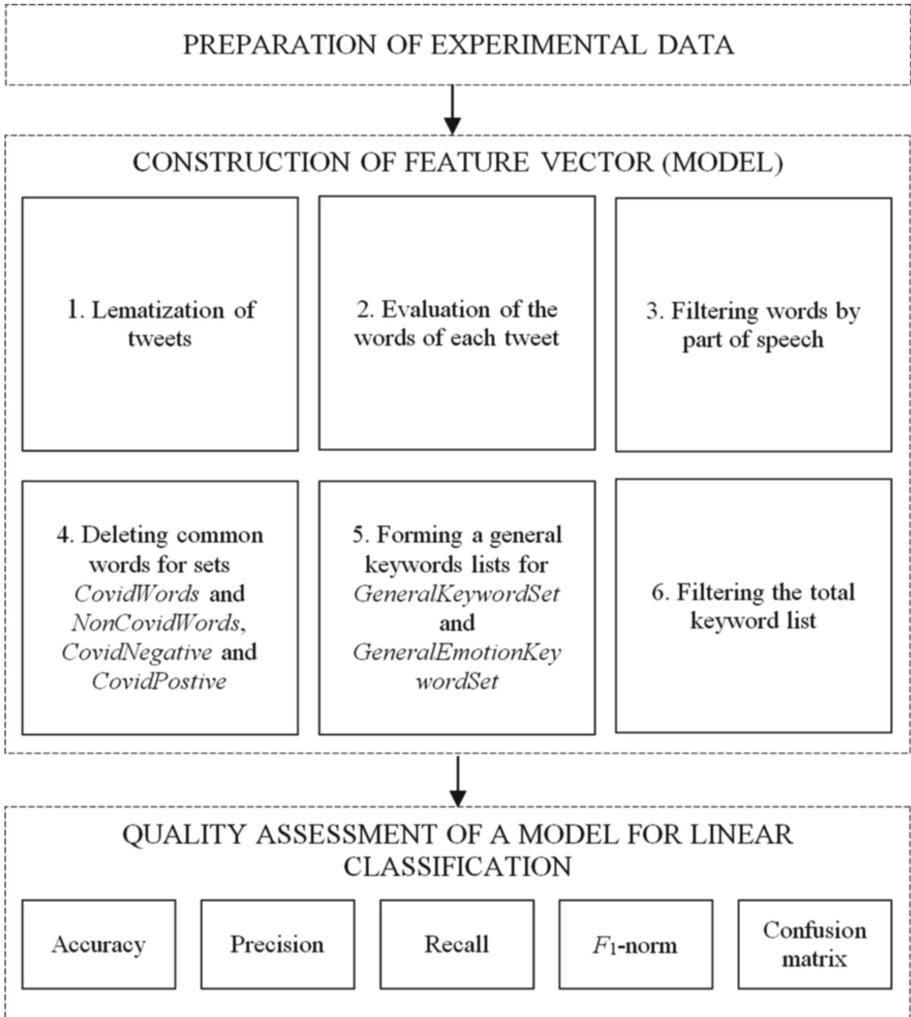


Fig. 1. Steps to build a vector of features (model)

- **Evaluation of the words of each tweet.** This stage is responsible for the formation of *CovidWords*, *NobCovidWords*, *CovidNegative* and *CovidPositive* set. Those sets contain evaluated words within the selected categories. The words of each tweet are evaluated by using the TextRank method.

Each element of the set is evaluated by the following formula

$$E_i = \frac{\sum_{j=0}^n tr E_{ij}}{n}, \quad (1)$$

where E_i is the evaluation of the i -th word in the set; trE_{ij} is the evaluation of the i -th word, which was obtained using the TextRank method within the j -th tweet; n - the number of tweets in the category.

This assessment provides the average importance of each word within the selected category.

- **Filtering words by part of speech.** The stage involves removing words from sets *CovidWords*, *NonCovidWords*, *CovidNegative* and *CovidPositive* which are not a noun, adjective, verb, or adverb. Words for which part of the language could not be determined are not deleted. Also, from sets *CovidWords* and *NonCovidWords* *deletes numbers* may be part of the content.
- **Removing common words** for sets *CovidWords* and *NonCovidWords*, and for sets *CovidNegative* and *CovidPositive*.

At first, it is needed to define the words in each of the sets.

$$CommonWords = CovidWords \cap NonCovidWords, \tag{2}$$

$$CommonEmotionWords = CovidNegative \cap CovidPositive. \tag{3}$$

Delete all words that match sets *CovidWords* and *NonCovidWords* as it will not be correct because for one of the sets such words can be essential. Therefore, deleting words from the set *CovidWords* which match the set *NonCovidWords* occurs by applying the following condition to the sets:

$$Ec_i < Enc_i \cdot \tau, i = 1...n, \tag{4}$$

where: *Enc* is the evaluation of the word in the *CovidWords* set; *Ec* is the evaluation of the word in the *NonCovidWords* set; n is the number of elements in the *CommonWords* set; τ is the fixed coefficient; the experimentally determined value was $\tau = 15$ for model “coronavirus” - “non-coronavirus”; by the same principle, $\tau = 5$ for model “coronavirus-positive emotion” - “coronavirus-negative emotion.” The following condition is used to remove words from the set *NonCovidWords* that match *CovidWords* from (4).

- **Forming general keywords sets** for *GeneralKeywordSet* for categories *CovidWords* and *NonCovidWords*, and *GeneralEmotionKeywordSet* for categories *CovidNegative* and *CovidPositive*.

Experimentally was found that it is enough to include the 400 elements from the *CovidWords* set into the *GeneralKeywordSet* set and near to 900 elements from the *NonCovidWords* set into the *GeneralEmotionKeywordSet* set for *CovidNegative* 250 elements and *CovidPositive* 250 elements. Words with the highest E_i score are selected from each sample.

- **Filtering the entire list of keywords.** In this step, the “expert” cleaning of the kits takes place *GeneralKeywordSet* and *GeneralEmotionKeywordSet* from possible incorrect values. In this case, incorrect meanings may defect words that do not convey meaning or are foreign. Such words include abbreviations, words from European languages, etc. As a result of cleaning the

set, *GeneralKeywordSet* consists of more than 1,200 words and the *GeneralEmotionKeywordSet* consists 400 words. The resulting *GeneralKeywordSet* set is a vector of features for vectorization of textual content into “coronavirus” and “non-coronavirus,” and the *GeneralEmotionKeywordSet* set for “coronavirus- negative” and “coronavirus-positive.”

Based on the received *GeneralKeywordSet* and *GeneralEmotionKeywordSet* vectorized content of tweets from the input set.

4.3 Assessment of the Classifier Quality

Assessing the quality of the proposed classifier is an essential step in the study. The following metrics are used to assess quality: accuracy, precision, recall, confusion matrix, and F_1 – norm [5,28].

The accuracy of the classifier is the proportion of correct predictions (both true positive and true negative) among the total number of cases considered, and precision indicates what proportion of the results found in the sample are relevant to the query. The accuracy of the classifier is determined by the formula:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

where TP is the true-positive predictions; TN is the true-negative predictions; FP is the false-positive predictions; FN is the false-negative predictions.

Precision is based on the formula:

$$accuracy = \frac{TP}{TP + FP}. \quad (6)$$

The best result is 1.0; it says that each of the samples in the input belongs to a particular class.

Recall shows the proportion of correctly classified documents that have been successfully found and is based on the formula:

$$recall = \frac{TP}{TP + FN}. \quad (7)$$

Measure F_1 is the weighted harmonic mean of precision and recall, which can be formally written as follows:

$$F_1 = 2 \times \frac{recall \times precision}{recall + precision}. \quad (8)$$

The best score for F_1 is 1.0, which suggests that precision and recall also have a value of 1.0. If F_1 is equal to 0, it means that either precision or recall is 0. If the set consists of unbalanced data, the measure F_1 may be inaccurate.

The confusion matrix shows the performance of the supervised learning algorithm. The rows of this matrix are samples of the predicted class; the columns are samples of the real class. The values of this matrix can be true positive, false positive, true negative, and false negative. The above metrics are used to analyze the results.

4.4 Experiment

The approach described above for the classification of user messages into “coronavirus” - “non-coronavirus” and “coronavirus-negative” - “coronavirus-positive” involves the division of the sample into educational and experimental. The most common ratio is 70/30%, but it has been experimentally established that a precise ratio is not crucial for the described approach.

Tweets taken from TBCOV and ukr-twi-corpus were filtered as follows

- The length of the messages is at least 100 characters.
- Although they did appear in the initial set, the coronavirus sample rejected messages that did not relate to the coronavirus topic.
- Messages written in a foreign language were deleted.
- Messages containing only emojis were also removed from the dataset. Thus, the dataset for classification as “coronavirus” - “non-coronavirus” consists of more than 2300 and 2900 tweets taken from the TBCOV and ukr-twi-corpus corpora, respectively. The dataset for classification as “coronavirus-negative” - to “coronavirus-positive” consists of more than 1100 positive emotions about coronavirus and 1100 negative. The scikit-learn library with the Python stack was used to conduct experiments [2]. Training and test datasets were obtained using the `train_test_split` method. The SVC classifier with a linear core is implemented using the `sklearn` library. Estimates of the classifier’s performance by metrics of classification precision, accuracy, recall, and confusion matrix were obtained with `sklearn` libraries.

5 Results and Discussion

The experimental dataset for classifying “coronavirus” and “non-coronavirus” messages consists of 2,214 Twitter messages. The 714 were related to the coronavirus, and 1,500 were non-related.

For visual analysis, the results of vectorization by the TextRank method were processed using MDS shown in Fig. 2.

As can be seen from Fig. 2, the proposed vectorization showed a linear resolution for both classification problems. The quality of the SVM classifier proposed in the article is investigated by the metrics described above. The test results for the classification “coronavirus” - to “non-coronavirus” are given in Table 1.

Table 1. Comparison of the use of metrics for the problem of classification “coronavirus” and “non-coronavirus”

Metrics	“coronavirus”	“non-coronavirus”
Precision	0.96	0.92
Recall	0.82	0.99
F_1 -score	0.88	0.95

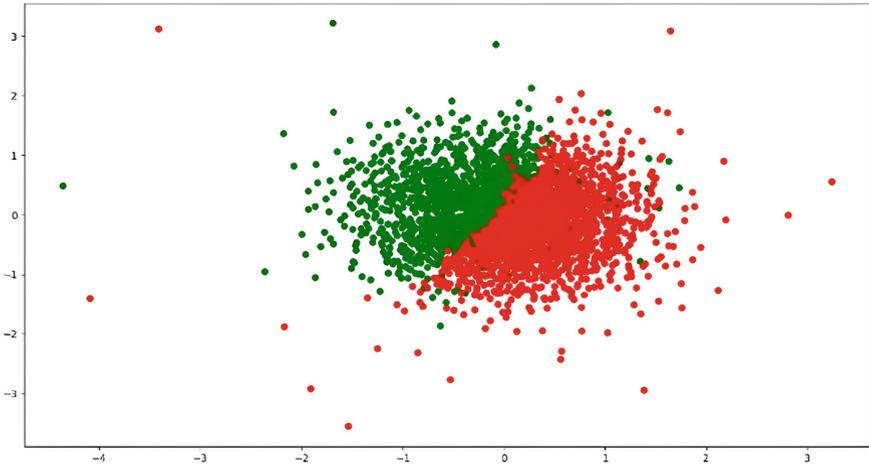


Fig. 2. Verification of the obtained vector of GeneralKeywordSet traits for the classification “coronavirus” - “non-coronavirus”

The assessment of the accuracy of the classification is 0.93. For class “coronavirus,” the precision value was 0.96; for second class “non-coronavirus” - 0.92 with a possible ideal value of 1.00. For class “coronavirus,” the recall value was 0.82; for second class “non-coronavirus” - 0.99 with a possible ideal value of 1.00. For class “coronavirus,” the -norm value was 0.88; for second class “non-coronavirus” - 0.95 with a possible ideal value of 1.00.

The confusion matrix that reflects the performance of the supervised learning algorithm proposed by the authors is presented in Table 2.

Table 2. The confusion matrix

Prediction	“coronavirus”	“non-coronavirus”
Predicted “coronavirus” class	582	132
Predicted “non-coronavirus” class	22	1478

As can be seen from the presented results, class “coronavirus” revealed 582 true-positive results, 132 false-positive, 22 false-negative results, and 1478 true-negative.

The 714 Twitter messages included in the coronavirus class were taken in the previous step to determine the sentiment of messages. Of these, 356 are coronavirus-negative, and 358 are coronavirus-positive. The sentiment classification was performed according to the same model as the previous classification task: “coronavirus-negative” and “coronavirus-positive,” i.e., on SVM with the keyword search method TextRank. The test results for classifying “coronavirus-negative” and “coronavirus-positive” are given in Table 3.

Table 3. Comparison of the use of metrics for the problem of classification “coronavirus-negative” and “coronavirus-positive”

Metrics	“coronavirus-negative”	“coronavirus-positive”
Precision	0.93	0.82
Recall	0.79	0.94
F_1 -score	0.85	0.87

The assessment of the accuracy of the classification is 0.86.

For class “coronavirus-negative,” the precision was 0.93; for second class “coronavirus-positive” - 0.82. Score by recall. For class “coronavirus-negative,” the recall value was 0.79; for second class “coronavirus-positive” - 0.94. For class “coronavirus-negative,” the -norm value was 0.85; for second class “coronavirus-positive” - 0.97.

The confusion matrix that reflects the performance of the supervised learning algorithm proposed by the authors is presented in Table 4.

Table 4. The confusion matrix

Prediction	“coronavirus-negative”	“coronavirus-positive”
Predicted “coronavirus-negative” class	280	76
Predicted “non-coronavirus-positive” class	22	336

As can be seen from the presented results, class “coronavirus-negative” revealed 280 true-positive results, 76 false-positive, 22 false-negative results, and 336 true-negative. The two classifiers have been combined to solve the problem of identifying sentiment messages on Twitter about the coronavirus pandemic. The following results of evaluating the classifier’s quality were obtained in Table 5.

Table 5. Comparison of the use of metrics for the problem of classification “coronavirus-negative,” “coronavirus-positive,” “non-coronavirus”

Metrics	“coronavirus-negative”	“coronavirus-positive”	“non-coronavirus”
Precision	0.88	0.83	0.92
Recall	0.64	0.80	0.99
F_1 -score	0.74	0.81	0.95

The assessment of the accuracy of the classification is 0.90. For class “corona-virus-negative,” the precision was 0.88, for class “coronavirus-positive”

- 0.93, and class “non-coronavirus” - 0.92. For class “coronavirus-negative,” the recall value was 0.64, for class “coronavirus-positive” - 0.80, and class “non-coronavirus” - 0.99. For class “coronavirus-negative,” the F_1 -norm value was 0.74, for class “coronavirus-positive” - 0.81, and class “non-coronavirus” - 0.95. Overall, the confusion matrix is presented in Table 6.

Table 6. The confusion matrix obtained by the proposed visual analytics-based method

Prediction	“coronavirus-negative”	“coronavirus-positive”	“non-coronavirus”
Predicted “coronavirus-negative” class	229	49	78
Predicted “coronavirus-positive” class	18	286	54
Predicted “non-coronavirus-positive” class	13	9	1478

The analysis revealed that some tweets identified as “coronavirus” in the sample were classified as “non-coronavirus”, for example:

- *Let’s scare you a little more :-) Comet C / 2019 Y4 continues to become brighter and the forecasts at the time of its flight near the Sun (end of May 2020) are too great, there are even those that predict brightness like the full moon!*
- *It would be good if public broadcasting similar to NHK in Japan appeared in Ukraine: with news on radio and TV, including the English version, a page with language lessons for all interested parties, etc.*

This is because these tweets are tied to a specific context. Even an expert can classify them as “non-coronavirus” if the tweet context is unknown.

There is also an error in the definition of emotions due to the peculiarities of the Ukrainian language. Thus, the following tweets, which in the initial sample were marked as negative, are classified as positive:

- *In Singapore, parliament, the president and the government will give up 1 month’s salary to give double the salary for 1 month to doctors and those who are fighting #coronavirus #Covid19*
- *This situation with the coronavirus has created the illusion that there are no diseases in the world other than this. We all know about our own, some are deadly, but we are afraid only of COVID-19*

Messages marked as positive but classified as negative:

- *National ingenuity. Translation - in the photo: creative ways that people have come up with to protect themselves from #coronavirus in China*
- *I plan to hold a marathon of films about epidemics. Outbreak and Contagion are next. Any other recommendations?*

6 Conclusions

The paper proposes the method for automated detection of short text messages' emotional sentiment, which allows automatically performing a multi-level classification of test samples of messages based on the set of available samples of short text messages. The method allows performing classification according to the content's semantic features and psycho-emotional features. In particular, an applied two-level classification was performed: in the first stage, the classification on the semantic basis "coronavirus" - "not coronavirus" (714 and 1500 tweets) was performed, and in the second stage, the classification on psycho-emotional content in the form of emotional coloring "negative" - "positive" (356 and 358 tweets). As a result, the model was obtained to classify short text messages according to the semantic and emotional features of the content in the form of the emotional coloring of the content "coronavirus-negative" - "coronavirus-positive." Studies conducted to classify 2214 short text messages from Twitter by semantic and emotional characteristics of the content revealed an average efficiency of classification of 90%. Resolved problem of identifying pandemic sentiment helps to track people's pandemic sentiments, their reactions to coronavirus-fighting messages, and local government decisions. Such data provides a feedback tool that can be used to quasi-dynamically monitor the situation and adjust the policy of anti-epidemiological measures. In the more extensive use, the proposed method provides the ability to target collective and individual behavioral anomalies for their timely blocking, which has significant social and political weight. Such behavioral anomalies that can be automatically tracked using the proposed method include trends in socially dangerous manifestations, acts of public disobedience, bullying, suicidal ideation, and other cases of aggressive, destructive influence.

Targeting sentiment messages on social media about the COVID-19 pandemic is a common problem for which many different approaches have been proposed. The peculiarity of the approach presented in this article is that it solves this problem for Ukrainian-language content, which has its peculiarities. Namely, the Ukrainian everyday language on social networks is inflected. It has numerous twisting words, surzhyk, and borrowing from other languages. The task of classifying Ukrainian language content is further complicated because there are no statistically verified corpora of the Ukrainian language. Studies show that the SVM-based classifier, using the TextRank method as a vectorization method, is high enough to consider the presented model effective in detecting coronavirus messages and further identifying positive and negative sentiment in Twitter messages. The approach described in this paper can be helpful in various government decisions to combat the coronavirus pandemic, such as increasing support for education and medicine. Further experimental investigations will be conducted to recognize and filter sensitive online content for psychologically sensitive people.

References

1. Large electronic dictionary of ukrainian (vesum). version 5.6.2 (2005–2022). <https://r2u.org.ua/vesum/>
2. Api reference (2022). <https://scikit-learn.org/stable/modules/classes.html>
3. Babichev, S., Lytvynenko, V., Skvor, J., Korobchynskiy, M., Voronenko, M.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction. In: 2018 IEEE Second International Conference on Data Stream Mining Processing (DSMP). IEEE (2018). <https://doi.org/10.1109/dsmp.2018.8478452>
4. Bobrovnyk, K.: Automated building and analysis of twitter corpus for toxic text detection. In: Proceedings of 3rd International Conference, COLINS 2019, pp. 55–56 (2019). <http://dx.doi.org/10.2139/ssrn.3123710>
5. Chakraborty, K., Bhatia, S., Bhattacharyya, S., Platos, J., Bag, R., Hassanien, A.E.: Sentiment analysis of COVID-19 tweets by deep learning classifiers—a study to show how popularity is affecting accuracy in social media. *Appl. Soft Comput.* **97** (2020). <https://doi.org/10.1016/j.asoc.2020.106754>
6. Chandra, M.A., Bedi, S.S.: Survey on SVM and their application in image classification. *Int. J. Inf. Technol.* **13**(5), 1–11 (2018). <https://doi.org/10.1007/s41870-017-0080-1>
7. Chintalapudi, N., Battineni, G., Amenta, F.: Sentimental analysis of COVID-19 tweets using deep learning models. *Infect. Dis. Rep.* **13**(2), 329–339 (2021). <https://doi.org/10.3390/idr13020032>
8. GitHub: Twitter scraper (2022). <https://github.com/bisguzar/twitter-scraper>
9. Hafidz, N., Liliana, D.Y.: Klasifikasi Sentimen pada Twitter Terhadap WHO Terkait COVID-19 Menggunakan SVM, N-Gram, PSO. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* **5**(2), 213–219 (2021). <https://doi.org/10.29207/resti.v5i2.2960>
10. Imran, M., Qazi, U., Ofli, F.: TBCOV: two billion multilingual COVID-19 tweets with sentiment, entity, geo, and gender labels. *Data* **7**(1), 8 (2022). <https://doi.org/10.3390/data7010008>
11. Kaur, H., Ahsaan, S.U., Alankar, B., Chang, V.: A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets. *Inf. Syst. Front.* **23**(6), 1417–1429 (2021). <https://doi.org/10.1007/s10796-021-10135-7>
12. Khan, R., Shrivastava, P., Kapoor, A., Mittal, A.: Social media analysis with AI: sentiment analysis techniques for the analysis of twitter COVID-19 data. *J. Crit. Rev.* **7**, 2761–2774 (2020)
13. Kovacs, B., Caplan, N., Grob, S., King, M.: Social networks and loneliness during the COVID-19 pandemic. *Socius Sociol. Res. Dyn. World* **7**, 1–16 (2021). <https://doi.org/10.1177/2378023120985254>
14. Krak, I., Barmak, O., Manziuk, E.: Using visual analytics to develop human and machine-centric models: a review of approaches and proposed information technology. *Comput. Intell.* **38**, 1–26 (2020). <https://doi.org/10.1111/coin.12289>
15. Krak, I., Barmak, O., Manziuk, E., Kulas, A.: Data classification based on the features reduction and piecewise linear separation. In: *Advances in Intelligent Systems and Computing*, vol. 1072, pp. 282–289. Springer, Heidelberg (2019). <https://doi.org/10.1007/978-3-030-33585-4-28>
16. Krak, Y., Barmak, O., Mazurets, O., and: The practice implementation of the information technology for automated definition of semantic terms sets in the content of educational materials. *Prob. Program.* (2-3), 245–254 (2018). <https://doi.org/10.15407/pp2018.02.245>

17. Kryvonos, I.G., Krak, I.V., Barmak, O.V., Bagriy, R.O.: New tools of alternative communication for persons with verbal communication disorders. *Cybern. Syst. Anal.* **52**(5), 665–673 (2016). <https://doi.org/10.1007/s10559-016-9869-3>
18. Kryvonos, I.G., Krak, I.V., Barmak, O.V., Bagriy, R.O.: Predictive text typing system for the Ukrainian language. *Cybern. Syst. Anal.* **53**(4), 495–502 (2017). <https://doi.org/10.1007/s10559-017-9951-5>
19. Lades, L.K., Laffan, K., Daly, M., Delaney, L.: Daily emotional well-being during the COVID-19 pandemic. *Brit. J. Health Psychol.* **25**(4), 902–911 (2020). <https://doi.org/10.1111/bjhp.12450>
20. Litvinenko, V., Burgher, J., Vyshemirskij, V., Sokolova, N.: Application of genetic algorithm for optimization gasoline fractions blending compounding. In: *Proceedings 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS 2002)*, pp. 391–394 (2002). <https://doi.org/10.1109/icaais.2002.1048134>
21. Lytvynenko, V., Savina, N., Krejci, J., Voronenko, M., Yakobchuk, M., Kryvoruchko, O.: Bayesian networks' development based on noisy MAX nodes for modeling investment processes in transport (2019). <http://ceur-ws.org/Vol-2386/paper1.pdf>
22. Lytvynenko, V., Lurie, I., Krejci, J., Voronenko, M., Savina, N., Taif, M.A.: Two step density-based object-inductive clustering algorithm. In: *Workshop Proceedings of the 8th International Conference on "Mathematics. Information Technologies. Education"* (MoMLeT and DS-2019), pp. 117–135. CEUR-WS, Shatsk, Ukraine (2019). <http://ceur-ws.org/Vol-2386/paper10.pdf>
23. Mansoor, M., Gurumurthy, K., U., A.R., Prasad, V.R.B.: Global sentiment analysis of covid-19 tweets over time (2020)
24. Prastyo, P.H., Sumi, A.S., Dian, A.W., Permanasari, A.E.: Tweets responding to the indonesian government's handling of COVID-19: sentiment analysis using SVM with normalized poly kernel. *J. Inf. Syst. Eng. Bus. Intell.* **6**(2), 112 (2020). <https://doi.org/10.20473/jisebi.6.2.112-122>
25. Radiuk, P.: Applying 3D U-Net architecture to the task of multi-organ segmentation in computed tomography. *Appl. Comput. Syst.* **25**(1), 43–50 (2020). <https://doi.org/10.2478/acss-2020-0005>
26. Radiuk, P., Barmak, O., Krak, I.: An approach to early diagnosis of pneumonia on individual radiographs based on the CNN information technology. *Open Bioinf. J.* **14**(1), 93–107 (2021). <https://doi.org/10.2174/1875036202114010093>
27. Radiuk, P., Kutucu, H.: Heuristic architecture search using network morphism for chest X-Ray classification. In: *1st International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS-2020)*, vol. 2623, pp. 1–15. CEUR-WS, Khmelnytskyi, Ukraine (2020). <http://ceur-ws.org/Vol-2623/paper1.pdf>
28. Rustam, F., Khalid, M., Aslam, W., Rupapara, V., Mehmood, A., Choi, G.S.: A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. *PLOS ONE* **16**(2), e0245909 (2021). <https://doi.org/10.1371/journal.pone.0245909>
29. Xue, J., et al.: Twitter discussions and emotions about the COVID-19 pandemic: machine learning approach. *J. Med. Internet Res.* **22**(11), e20550 (2020). <https://doi.org/10.2196/20550>

30. Zhang, M., Li, X., Yue, S., Yang, L.: An empirical study of TextRank for keyword extraction. *IEEE Access* **8**, 178849–178858 (2020). <https://doi.org/10.1109/access.2020.3027567>
31. Zhou, N., Shi, W., Liang, R., Zhong, N.: TextRank keyword extraction algorithm using word vector clustering based on rough data-deduction. *Comput. Intell. Neurosci.* **2022**(4), 1–19 (2022). <https://doi.org/10.1155/2022/5649994>



Software Based on Ontological Tasks Models

Yevhen Burov¹(✉) , Victoria Vysotska^{1,2} , Vasyl Lytvyn¹ ,
and Lyubomyr Chyrun³ 

¹ Lviv Polytechnic National University, Lviv, Ukraine

{yevhen.v.burov,victoria.a.vysotska,vasyl.v.lytvyn}@lpnu.ua

² Osnabrück University, Osnabrück, Germany

victoria.vysotska@uni-osnabrueck.de

³ Ivan Franko National University of Lviv, Lviv, Ukraine

Lyubomyr.Chyrun@lnu.edu.ua

Abstract. The growing complexity of software reflects the trend of mobility and changeability of business processes in our increasingly interconnected world. The change in the business process entails the difference in the supporting software, which should be updated quickly and efficiently. Existing software architectures and approaches to design cannot provide a quick software adaptation to the change in requirements. This article explores using task ontologies as building blocks for software. Such software is modeled as a network of executable task models. Each model can initiate actions such as executing operating system commands, querying a database or sending requests to external services. While the task model can perform a small task, the network of interacting models can perform complex tasks. The article describes the application of the proposed approach in automated software testing. The advantages of using task models in software are analyzed for different types of change requirements compared to the traditional software design approach. The prototype of the modeling tool, allowing the creation of onto-logical task models and their execution, is described.

Keywords: Ontology · Ontological model · Software design · Adaptation · Business requirements

1 Introduction

The analysis of trends in software systems architectures shows that their complexity and the pace of updates continue to increase. It is in line with the general trends of globalization in the world economy and the need to respond quickly to changes in the business environment. The increasing mobility of business processes implies the widespread use of software systems and technologies and demands their ability to adapt quickly and accurately to changes. The existing software architectures and approaches to its design do not fully address this

problem. Consequently [13,15], the part of IT budgets spent on software support is increasing (from 67% in 1979 to 90% in 2000). Lehmann at first studied this problem in the framework of the theory of software systems evolution [16]. The researcher identified three classes of systems (P, S and E) and formulated the laws of software systems evolution. In particular, for class E systems (which correspond to the processes of the outside world and must constantly reflect the state of the environment), it is stated (Laws 2 and 7) that over time the complexity increases and the quality of such systems decreases. The high level of complexity of software system modification resides in the number of flaws and bottlenecks inherent to existing architectures of software and methods of its construction. In particular, in such systems, the rules and algorithms of operations are rigidly embedded in the code; there is a separation of the business process system from the software system that ensures its functioning; the design of a software system is based on a fixed set of system requirements, which are often not fully understood. The increasing degree of integration of business processes in enterprises, the development of e-commerce and the globalization of the world economy lead to an increase in the pace of change in the business environment and, consequently, the need for constant adaptation of software to the changing environment. The traditional design methods and architectural software development principles focus on creating a software product based on a fixed set of requirements. As a result, we obtain the systems that respond poorly to changes in requirements and external factors, are expensive for support and operation, and often have bugs and flaws. The update of software to changes in requirements usually leads to the need to release a new version, which involves performing long and resource-intensive stages of analysis, design, coding, testing and implementation of the new version of the product. Such update often stems from the changes in the business process, which requires the knowledge of experts in the relevant subject area. As a rule, such experts are not specialists in programming. Hence the need for a clear formulation of requirements by experts for system developers. Errors and misunderstandings in managing system requirements during conceptualization make up a large part of the causes of software development failures [5]. Thus, there is a need to develop new architectural principles, methods, and tools to simplify the construction and modification of software systems and enhance their adaptability to changes in requirements. The main contributions of this paper are summarized as follows:

1. The paper investigates the problem of increasing software complexity in the process of its evolution and proposes to use task ontologies as software building block to counteract this problem
2. The formal model of task ontologies, incorporating the set of actions and based on algebraic approach was developed.
3. In order to model the solution of complex problems, including the execution of sequences of interrelated tasks, the mechanism of task models interaction was proposed.

4. The vision of software as multi-level system performing tasks on different abstraction levels and built using corresponding conceptual task models was developed.
5. The prototype of task-based software environment was developed and described.
6. The advantages of software built using task models compared to traditional software development were investigated.

The paper is structured in such way. Section 2 contains comparison of proposed approach to existing model-based software architecture. Section 3 presents the details and formal model of task-based approach. Section 4 has examples, the description of modeling software and discussion about advantages of the pro-posed approach to software construction. In conclusion, we summarize the results.

2 Related Works

Several scientific and technological efforts are aimed at solving the problem mentioned above. One way to solve it is to separate the logic of the software system from the mechanism of its processing and implementation. In this approach, the logic of operation is presented in the form of a certain formal model. An example of such approach implementation is MDD/MDA (Model-driven design/model-driven architecture), created by Shlaer and Mellor. The software logic is specified as the formal model and then compiled into code. The usage of this approach and its advantages revealed its significant disadvantages, particularly the complexity of creating and modifying a set of models, comparable to the complexity of traditionally creating a system. Another way to implement MDD is to use ontologies to build software systems. Indeed, by Gruber's definition [10], an ontology is a formal model of conceptualization of the subject domain. This model contains definitions of domain entities and dependencies between them. When building ontology-based software systems, developers avoid re-conceptualizing the subject area, reducing the use of resources at the system analysis and design stages. The obstacle to the use of ontologies in the construction of software systems is the purely declarative nature of ontologies and the lack of presentation of procedural knowledge in them. On the other hand, in software development methodologies, the impact of unclear requirements is countered by the use of Agile programming methods. The software is developed as a sequence of short projects with clear, actionable requirements. Before starting each new project, the requirements are re-analyzed and refined [3]. At the same time, such development methodologies are effective only in creating relatively simple software. The generally accepted practice of considering the business requirements is modeling business processes and using its results to build the software. To take into account the specifics of business processes, a level of business logic (Microsoft .NET WPF application framework, Domain driven design [9]) was introduced into the architecture of the software. At the same time, such implementations are usually static and take

into account only the state of business rules at a certain point in time. Implementing business rules as part of the software code significantly complicates its modification. The research of Ross, Hay and several other authors [4, 23] propose a declarative description of business rules in a separate specialized system. This part of the software is the rule engine that monitors the implementation of business rules. This approach allows adjusting the system by changing business rules flexibly [19]. Its limitation comes from the inability to present all the business environment features in the form of rules, the lack of a declarative description of the environment, and the tracking dependencies difficulty between rules. A promising direction in solving the problem of adapting software to the changes in the environment is using approaches of knowledge engineering, particularly ontological modeling [18]. Unlike modeling approaches that focus on compiled models or rule processing, the ontological modeling approach builds a formal domain model (ontology), considering its features and limitations. The same model can be reused to create other software systems for the same subject area. The summary of the analysis of known approaches in the context of domain modeling in software architectures is given in Table 1.

Table 1. Domain modeling in software architectures

Architecture	Domain modeling	Disadvantages
Client-server, DCOM, CORBA, Java	Explicit modeling is not performed	Business logic and data implicitly include code, which complicates its modification
NET, WPF, Common application architecture, Domain-driven architecture	Building a separate business model and, based on it - software business components and services	Business components reflect the state of the environment at one point in time
SOA	Building a business model and analyzing the strategy and trends of the business environment	The complexity of implementing unforeseen changes
MDA	Based on the model specification of the subject area, build a specification of the software system, which is compiled into code	Considerable complexity of model specification development
Rule-based architectures	The environment model, represented as rules, is part of the soft-ware system	Absence of a coherent model of the environment, the complexity of rules harmonization
Ontology-based on architectures	The domain model is represented by an ontology and is holistic	The ontology construction and maintenance complexity

3 Materials and Methods

3.1 The Use of Task Ontologies in Software Design

To build software systems using an ontological approach, we focus on tasks which are the smallest and most necessary part of any process. The task notion is defined [2] in the literature as a smallest identifiable and essential piece of a job that serves as a unit of work. Task ontologies were first mentioned as a result of research in task analysis. Task analysis methods are used to identify and formalize all the factors that affect the process of solving a problem by an expert. Such methods are widely used for designing computer program interfaces, expert systems, and decision support systems. The main effort in this area is to identify the components of the problem, its structure and limitations. It allows the expert to understand it better, model the problem, evaluate the results and transfer their knowledge to other experts [12]. The area of task analysis has undergone significant changes with the advent of ontological modeling. It has been suggested to use task ontologies to formalize task concepts and relationships [25]. Unlike other ontologies, such as upper or domain ontologies, task ontologies are built separately for groups of similar tasks, introducing the notion of action [22] and providing the means of task execution or modeling. Task ontologies research is closely related to conceptual modeling because, in constructing a task ontology, it creates its formalized conceptual model. An essential aspect of conceptual and ontological task modeling is the interaction with the subject matter expert who creates and validates the ontology. In ontology research, simulation environments have been implemented that allow the creation and execution of ontology models for specific task classes [17]. The most advanced class of such environments is CLEPE (Conceptual level programming environment) [11]. Recently task models were used in robotics applications [24]. However, there is difficulty in transitioning from conceptual model to executable code. An approach to the construction of software systems using interpreted ontological models is proposed along with the architecture of the modeling system. At the same time, the aspect of mathematical formalization of the proposed approach, particularly the presentation of knowledge using ontologies and models, the dependencies between them, and the question of practical implementation of the modeling system, requires additional research. This work aims to present developed architectural principles, methods, and tools for building software systems based on ontological task models.

3.2 Formal Representation of Knowledge and Modeling System Structure

The use of ontological models for the construction of software systems requires the development of a mathematical apparatus in the form of a suitable model, which will later be used for the formal specification of modeling methods and techniques. An algebra based approach was used to construct the formal model, which determines the multi-sort algebraic system by combining several algebraic domains. Let there be n sets of objects in the subject area A_1, A_2, \dots, A_n . The

objects belonging to each set are classified as instances of a particular concept. These sets are carrier sets for n multi-sort algebras. We denote the individual instances belonging to these sets as a_1, \dots, a_n . Let's define the domains of multi-sort algebra.

Concept Domains (Entities) E . Based on each set A_i , define an abstract data type E_i .

$$E_i = (Name, \Sigma, Ex), \tag{1}$$

where $Name$ is the name of the type, Σ is the signature of an algebra, Ex are the constraints. We will denote individual types as E_i (arbitrary type, its name is not defined) or where name is the type name if it is important to refer to a specific type in the consideration context. We denote variables that accept values of instances of a particular type. Types form an algebraic domain E that correspond to the concepts (entities) of the domain: $E = E_1, E_2, \dots, E_n$.

Attribute Domain At . We define the algebraic domain of the attributes At as the list of attribute values in the form of pairs ($key, value$). The key pair element defines the attribute ID and $value$ - its value. The operations merge, substitute, delete, and interpretation is defined for this domain. In practice, functions that are defined on variables of different domains are useful, for example, the function of choosing the attribute value:

$$F_{selval}(At, key) \rightarrow value. \tag{2}$$

Boolean domain Cs . The Boolean domain includes expressions whose result is a *Boolean* value $\{true, false\}$. Expression operands are variables belonging to other algebraic domains. We will interpret the elements of the Boolean domain as a constraint type Cs . The operations for the *Boolean* domain are Boolean operations and, or, negation, as well as the interpretation operation, which reflects the simplification and calculation of Boolean expressions. An instance of a Boolean domain is a specific constraint.

The Entity with Attributes Domain Et . It is defined on the set of tuples $\{(E_i, At_j, Cs_j^*)\}$. For each i , there is only one j that is part of this domain:

$$\forall i \exists^1 j : (E_i, At_j). \tag{3}$$

Each constraint $Cs_j \in Cs_j^*$ is an expression with operands from the domain At_j . Operations on elements of this domain are $\{merge, split\}$. An entity *merge* operation is interpreted as forming a new entity as a common set of properties and constraints on entities. The entity *split* operation is the reverse of the merge. An instance of an entity domain with attributes is a tuple describing a fact.

The Domain of Relations Rl . The carrier set of this domain is formed with such structures:

$$\{(Et_{1,i} \times Et_{2,i} \times \dots \times Et_{k,i}, At_i^r, Cs_i^*)\}. \tag{4}$$

Each structure is a tuple containing Cartesian product of algebraic data types from domain T , a type from attribute domain At (defines the attributes of relation), and a set of constraints Cs . Each constraint $Cs_i \in Cs_i^*$ is a Boolean expression with domain-operands $At_{1,i}, At_{2,i}, \dots, At_{k,i}, At_i^r$. For relations above are defined the operations of union, splitting, substitution $\{merge, split, substitute\}$. Merge and split operations are treated similarly to operations in domain E . We treat a relation as an entity with attributes in a substitution operation.

An Ontology On is a tuple represented by domains of entities with attributes, relations and corresponding constraints from the Boolean domain:

$$On = (Et, Rl, Cs^*). \quad (5)$$

Let's define a **task ontology** On_{TS} which is a part of the domain ontology On and contains elements needed for the execution of specified task:

$$On_{TS} \subseteq On \quad (6)$$

Ontological Task Model Md is defined as a tuple with three elements:

$$Md = (On_{TS}, Ac_{TS}^*, Cs_{TS}^*) \quad (7)$$

where On_{TS} is the task ontology, Ac_{TS}^* is the set of actions, Cs_{TS}^* is the additional constraints set. Action Ac_{TS} is an entity with attributes interpreted as a command to some external service or instruction to activate some other model. Table 2 shows the basic algebraic domains used in mathematical formalization. The analysis of the knowledge representation and processing methods shows that most of them are based on a declarative approach. In particular, this category includes frame logic systems, ontological systems based on OWL and others. The advantage of a declarative approach for knowledge representation resides in the ability to create a coherent formal do-main model and use powerful mechanisms of logical inference. Its disadvantages include the significant complexity of ontology management processes and the lack of representation of procedural abstractions. An alternative to the declarative approach to knowledge representation is a procedural approach, particularly SBD (Schema-based design) [21]. The advantages of the procedural process are the ease of creating and using schemas, the similarity of schemas to mental models, and task orientation. The disadvantages of the procedural approach are the poor coherence and consistency of different schemes, the lack of a holistic representation of the subject area and the tools for storing and processing the schemas.

Given the complementary nature of declarative and procedural approaches, it is appropriate to combine them. The ontology defines a set of entities, relations, axioms and constraints for the entire subject area. Ontological task models play the role of schemas. They use the ontology components and define the entities, relationships, constraints and actions in the context of a particular task. Figure 1 shows the structure of a software system based on ontological problem models. The central element of the system is a knowledge base that contains

Table 2. Models of algebraic domains

Domain	The carrier set	Operations
Entities E	$\{a_i Type(a_i) = E_i\}$	
Attributes At	$\{(key, value)^*\}$	merge, substitute, delete, interpret
Boolean domain Cs	$\{true, false\}$	and, or, negate, interpret
Entities with attributes Et	$\{(E_i, At_j, Cs_j^*)\}$	merge, split
Relations Rl	$\{(Et_{1,i} \times Et_{2,i} \times \dots \times Et_{k,i}, At_i^r, Cs_i^*)\}$	merge, split, substitute
Ontology On	$On = (Et, Rl, Cs^*)$	merge, split
Ontological models Md	$(On_{TS}, Ac_{TS}^*, Cs_{TS}^*)$	activate, deactivate

an ontology, a fact base and a set of models. The fact base stores facts about objects and events of the outside world that are instances of classes defined by an ontology. The ontology contains a domain model, presented as a taxonomy of classes. It creates the ability to uniquely interpret all objects from the fact base and elements of ontological models. Ontological models reflect knowledge of how to execute specific tasks. Models form fact models based on models - they are models initialized by particular facts from the fact base. Models perform actions by developing commands and queries to external services. Such services include operating system services, SOA compliant enterprise information system services, or arbitrary web services.

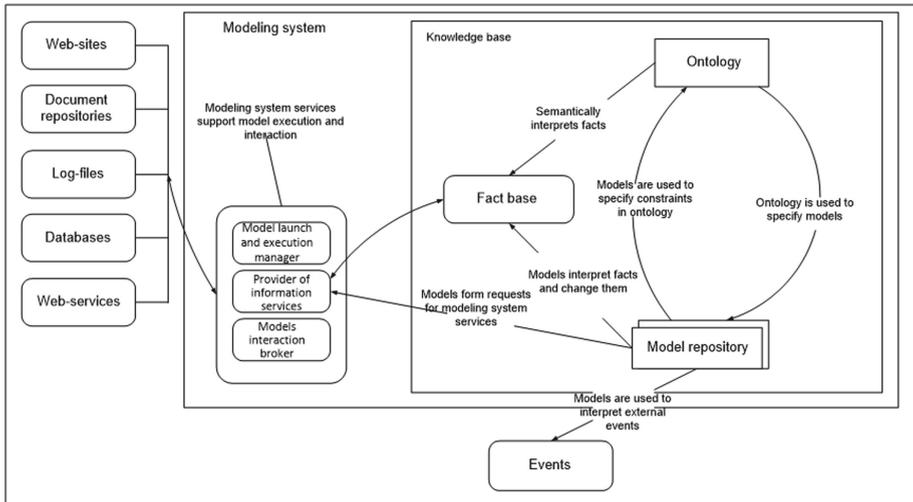


Fig. 1. The conceptual structure of software based on task ontologies

3.3 Mathematical Model of Representation and Processing of Knowledge in the Modeling System

Let's define an ontology On as the tuple of entities Et , relations between them Rl and constraints Cs :

$$On = (Et, Rl, Cs), \tag{8}$$

Each relation $Rl_i \subseteq Rl$ is defined in multiple roles $\{P_1, P_2, \dots, P_n\}$. In the general case, an F_{in}^k initialization function is defined for each role P_k , which specifies a subset of entities whose elements are allowed to initialize the role:

$$F_{in}^k : P_k \rightarrow Et_{in}^k \subseteq Et, \tag{9}$$

In the simplest case, when $\forall k : |Et_{in}^k| = 1$ there is only one type of entity for each role. In this case, we can replace roles with the relevant ontology entities: $Rl(Et_1, Et_2, \dots, Et_n)$. Ontology entities form a hierarchical structure (taxonomy) using the subsumption relation Rl_{isa} .

To represent the modeling system formally, we define an algebraic system of abstract data types \bar{T} in which for each element T there is a one-to-one correspondence with a particular entity of ontology:

$$\forall i \exists^1 j : T_i \rightarrow Et_j \ \& \ \forall j \exists^1 i : Et_j \rightarrow T_i, \tag{10}$$

The $TypeEn()$ function returns for each data type T_i corresponding ontology entity Et_j

$$TypeEn(T_i) = Et_j, \tag{11}$$

and thus, determines the semantic interpretation of this type of data. According to the approach [14] abstract data type is specified by the tuple

$$T = (Name, \Sigma, Ex), \tag{12}$$

where $Name$ is the name of the type, Σ is the signature of a multi-sorted algebra, Ex is the set of equations in the signature Σ that specify the definitions of the abstract data type. The tuple specifies the signature

$$\Sigma = (S, OP), \tag{13}$$

where S is the set of the base set of names and OP is the set of operation names: $S = \{S_1, S_2, \dots, S_n\}$, $OP = \{F_1, F_2, \dots, F_k\}$. Each operation Op_i defines a mapping

$$Op_i : S_{a(1,i)} \times \dots \times S_{a(n,i)} \rightarrow S_{m(i)}. \tag{14}$$

The relation Rl can be considered as a kind of algebraic operation that acts on a given set of argument types and defines the mapping to a Boolean set

$$Rl_i : S_{a(1,i)} \times \dots \times S_{a(n,i)} \rightarrow Cs. \tag{15}$$

The model Md can also be considered as a kind of operation that uses the specific set of argument types and defines a set of results, which in general have different types:

$$Md_i : S_{a(1,i)} \times \dots \times S_{a(n,i)} \rightarrow \{S_{r(1,i)}, \dots, S_{r(l,i)}\}. \tag{16}$$

On the other hand, the ontology of relations and models is a separate entity. These entities in the algebraic type system \bar{T} correspond to the types of data whose instances store data about these relations and models (metadata). An important part of the definition of the relation is the integrity constraints imposed on the entities combined by the relation. In a particular type of system, such a constraint is represented by a set of Boolean expressions $Cs_{T_{Ri}}$ for each relationship type T_{Ri} , which should hold

$$\forall cs_{T_{Ri}}^i \in Cs_{T_{Ri}} : ev(Cs_{T_{Ri}}^i) = true, \quad (17)$$

where $ev()$ is the function of evaluating the expression value.

Expressions that define integrity constraints are part of the set of defining relationships of an abstract data type

$$Cs_{T_{Ri}} \subseteq Ex_{T_{Ri}}. \quad (18)$$

3.4 The Model of Ontological Task Models Interactions

Unlike ontology classes, models do not form a clear hierarchy and create a dynamic network in which relations and the models themselves can change, reflecting learning processes, changes in the outside world, or executing a specific task. Each model can be in one of two states - active or passive. Accordingly, we divide all instances of models into subsets of active and passive models that do not intersect:

$$Population(T_{MD}) = \hat{t}_{MD}^{ac} \cup \hat{t}_{MD}^{ps}, \hat{t}_{MD}^{ac} \cap \hat{t}_{MD}^{ps} = \emptyset, \quad (19)$$

where $\hat{t}_{MD}^{ac}, \hat{t}_{MD}^{ps}$ are sets of instances of active and passive models.

An active model is a model initialized by contextual information. Models go into the active state at the request of other models or when certain events occur. Dynamic models are used to execute the current tasks and interpret and update the knowledge in the system. If the active model is no longer needed (result is obtained, goal is reached), the model goes from an active to a passive state. Model interaction T_{RMD} is a type of data that reflects the activation relationship that is used to make the decision to activate the model (s). Let's look at the process of model interaction and activation in more detail. The initiator of communication between the models is the actuator model. Model interaction arises when the currently executed models need to solve auxiliary problems. For example, when initial data are incomplete, the model needs to obtain more data for execution. Such a decision may lead to searching for the information in a database, and consulting a human expert on the Internet. Each sub-type of T_{RMD} - T_{RMD}^i exists a class of T_{PR}^j tasks which can be activated. This class of tasks corresponds to the set of models that can be applied to solve the problem of this class. To initialize a model, the problems of determining relevance, optimal choice among relevant models and initialization of the selected model should be resolved. The relevancy function performs a mapping from the current context t_{CON} and the

set of alternatives \hat{t}_{MD} to the set $\{true, false\}$, defined as a result of the subset of applicable models $\hat{t}_{MD}^{re} \subseteq t_{MD}$:

$$F_{rel} : (t_{CON}, \hat{t}_{MD}) \rightarrow \{true, false\} \tag{20}$$

In the absence of relevant models, the modeling system sends a message that the problem cannot be solved to the activator. The choice function selects from relevant models one, which maximizes the function of choice t_{MD}^{op} , taking into account selection criteria \hat{t}_{CR} and context t_{CON} :

$$F_{ch}(t_{MD}^{op}, \hat{t}_{CR}, t_{CON}) \rightarrow max \tag{21}$$

The initialization function F_{in} maps the current context t_{CON} to the set of attribute values of the selected model - t_{VSL} :

$$F_{in} : t_{CON} \rightarrow t_{VSL} \tag{22}$$

To summarize, let us define T_{RMD} as a tuple:

$$T_{RMD} = (T_{PR}, \hat{T}_{MD}, F_{rel}, F_{ch}, F_{in}). \tag{23}$$

The model type T_{MD} consists of schema types T_{SCM} and implementations T_{IMD} :

$$T_{MD} = (T_{SCM}, T_{IMD}). \tag{24}$$

The model schema describes its structure, defines rules and constraints for using the model, and a list of possible operations and requests. The schema is a model component that is visible to the outside world. It is used to interact with the model. At the level of ontology, a model is defined as a tuple of sets of operations $E\hat{t}_{op}$, entities $\hat{E}t$ relations $\hat{R}l$, additional constraints, $\hat{C}s$ and rules $\hat{R}u$:

$$E_{MD} = (E\hat{t}_{op}, \hat{E}t, \hat{R}l, \hat{C}s, \hat{R}u). \tag{25}$$

The model schema consists of slots T_{SIM} , their relations \hat{T}_{RSM} , rules \hat{T}_{RUM} , constraints \hat{T}_{CSM} and operations \hat{T}_{OPM} :

$$T_{SCM} = (\hat{T}_{RSM}, \hat{T}_{RUM}, \hat{T}_{CSM}, \hat{T}_{OPM}). \tag{26}$$

The model slot is a role attribute. For each slot, a function F_{RG} is specified that defines the set of classes \hat{T}_{CL}^{RG} whose objects are allowed to initialize the slot:

$$F_{RG} : T_{SLM} \rightarrow \hat{T}_{CL}^{RG} \tag{27}$$

The relation of slots T_{RESM} is specified by the slots it connects \hat{T}_{SLM}^{RESM} by the set of ontology classes used to interpret the relation semantically - \hat{T}_{SL}^{RESM} and by the set of models used to understand and perform relation operations - \hat{T}_{MD}^{RESM} :

$$T_{RESM} = (\hat{T}_{SLM}^{RESM}, \hat{T}_{SL}^{RESM}, \hat{T}_{MD}^{RESM}). \tag{28}$$

the slots connected by relation belong to model slots:

$$\hat{t}_{SLM}^{RESM} \subseteq Population(T_{SLM}). \quad (29)$$

Model relations correspond to one of the types of relations defined in the ontology On :

$$TypeEn(T_{RESM}) = Rl_{RESM} \in Rl. \quad (30)$$

The model describing the relations is an element of the set of models:

$$t_{MD}^{RESM} \in Population(T_{MD}). \quad (31)$$

Let t'_{BFC} be a certain situation, a state, a snapshot of the fact base. We define the goal data type T_{GL} with instances specifying the states of the fact base when the goal is achieved:

$$Population(T_{GL}) = \hat{t}_{BFC}^{GL} \subseteq Population(T_{BFC}). \quad (32)$$

Working with goals, it is useful to specify a goal function that allows for determining whether a goal Gl is achieved in a particular state t'_{BFC} :

$$F_{GL}(t'_{BFC}) = \begin{cases} true & | t'_{BFC} \in \hat{t}_{BFC}^{GL} \\ false & | t'_{BFC} \notin \hat{t}_{BFC}^{GL} \end{cases} \quad (33)$$

Such goal function can be specified, for example, as an ordered list of assertions \hat{t}_{ASR} that can be checked for the current state of the fact base:

$$F_{GL}(t'_{BFC}) = \hat{t}_{ASR}(t'_{BFC}), \quad (34)$$

where $\hat{t}_{ASR}(t'_{BFC})$ is the assertion requirement for the values of the properties of objects and their relations in the state t'_{BFC} . Each statement t'_{BFC} is a function specified on the set $\{true, false\}$:

$$Range(\hat{t}_{ASR}(t'_{BFC})) = \{true, false\}, \quad (35)$$

where $Range(f)$ function specifies the value range of the function f . Each ontological model is intended to execute the specific task for the specified goal. For the convenience of finding the relevant models, it is advisable to organize the information about the models according to the classes of problems they solve via ontology $On_{BL} \subseteq On$. For example, we would be able to separately define classification models, algorithmic models, service models, access control models, and situational models. The Model Execution Manager service processes model instances. Model interaction broker uses a goal ontology to find the model needed to solve the specified problem. The formal definition of context is necessary to ensure model interaction. Models share relatively small amounts of information, providing only links to specific facts. The model must obtain all the necessary information to operate on its own in the context of these facts. While doing so, it is often following the chains of facts linked by relations. The formal definition of

context is given at the level of ontology (context of knowledge) for a specific class and the level of facts in a particular fact. A central element of the knowledge context is a particular class T^{co} . The context of zero-level knowledge is the class itself and the empty set of its connections:

$$Con_{kn}^0 = (S_{co}^0(T_{co}), S_{co}^0(T_{RCL}^{co})) = (T^{co}, \emptyset). \quad (36)$$

The first-level knowledge context contains all the relationships and classes that the context class directly connects to:

$$\begin{aligned} Con_{kn}^1 &= Con_{kn}^0 \cup (S_{co}^1(T_{co}), S_{co}^1(T_{RCL}^{co})), \\ \forall T^i \in S_{co}^1(T_{co}) \exists T_{RCL}^j &: T^{co} \in T_{CL-RCL}^j, T^i \in T_{CL-RCL}^j \\ \forall (T^i, T^j) \mid (T_{RCL}^j, T_{RCL}^i) &\in S_{co}^1(T_{RCL}^{co}) : T^i \in S_{co}^0(T_{co}), T^j \in S_{co}^1(T_{co}). \end{aligned} \quad (37)$$

similarly, the context of the level k is:

$$\begin{aligned} Con_{kn}^k &= Con_{kn}^{k-1} \cup (S_{co}^k(T_{co}), S_{co}^k(T_{RCL}^{co})), \\ \forall T^i \in S_{co}^k(T_{co}) \exists (T_{RCL}^j &: T^l \in S_{co}^{k-1}(T^{co}) : T^i \in T_{CL-RCL}^j, \\ &T^l \in T_{CL-RCL}^j, \\ \forall (T^i, T^j) \mid (T_{RCL}^j, T_{RCL}^i) &\in S_{co}^k(T_{RCL}^{co}) : T^i \in S_{co}^{k-1}(T_{co}), T^j \in S_{co}^k(T_{co}). \end{aligned} \quad (38)$$

In general, the context of T^{co} is the maximum context level m that can be built in a given knowledge base:

$$Con_{kn}(T^{co}) = Con_{kn}^m(T^{co}) \mid m = \max(k). \quad (39)$$

4 Experiment, Results and Discussion

4.1 Software as a Multi-leveled Hybrid System Based on Ontological Task Models

The disadvantages of existing model-oriented approaches for software construction stem from the complexity of both the subject area and relevant models. We suggest counteracting this flaw by implementing software as simple, interacting and interpreted task models. The executable and interpretable nature of models creates several additional advantages over traditional software development methods and xUML, in particular:

- There is no need to recompile and re-deploy the system when business logic has changed. Only models implementing this logic need to be modified;
- The use of ontological models enables the accumulation and reuse of knowledge about how to solve problems

The accumulation and reuse of knowledge in software are accomplished by creating and using ontological models of tasks at three conceptual levels: business

process level, service and application level, and device level (Fig. 2). In particular, business process models and business logic are defined at the business process level. The models use concepts that reflect the characteristics of business processes. In case of business process changes, these changes are expressed as a change in requirements to service and application-level models and passed down to this level. We consider the applications and system-wide services and resources used by business processes at the service and application level. It considers resource sharing, formulates and implements system-wide corporate policies, and prioritizes tasks. Device level contains models of task processors. Such processors can be both machines and humans for different types of tasks. At this level, the modeling system works with parameters, limitations, and operation models of processors and strives to maximize task execution efficiency. Therefore, the software is considered as a tuple of sets of models:

$$PS = (S_{MD-BP}, S_{MD-SE}, S_{MD-PC}), \quad (40)$$

where S_{MD-BP} is the set of business process models, S_{MD-SE} is the set of service models, S_{MD-PC} is the set of device models. At each level, a set of interconnected models are used to implement adaptation tasks within that level. Thus, at the level of business processes, we can adapt the system's behavior by changing the priority of tasks or moving to another variant of the process structure. One of the adaptation solutions is to change the process level requirements or change the device settings. The choice an adaptation approach is chosen within the level based on the available knowledge, depending on the characteristics of the situation that requires adaptation. At the level of business processes, the focus is on that part of the overall business model that is implemented using computer technology information technologies [20]. Depending on their generality level, we distinguish between general and specific models. General models represent knowledge in many subject areas (they should be based on upper-level ontologies). Examples of such general models are time, personality, document, and organizational structure models. Specific models are used to represent knowledge in a particular subject area. An example is such a model in the software development process model. Another classification of models can be provided by the classes of problems they solve and the features of the model architecture: algorithmic, situational models, classification models, operation models, etc. As a rule, all models in one class refer to one common model that presents the model task in its most general form.

4.2 The Use of Ontological Task Models in Software. Application Example

Simple implementations of the proposed approach were developed as a proof of concept in subject areas of software testing and access control in document management. As a basis for analysis of processes and tasks of the software testing, we have taken the processes defined in ISO/IEC/IEEE 29119 [1]. This standard defines three groups of processes:

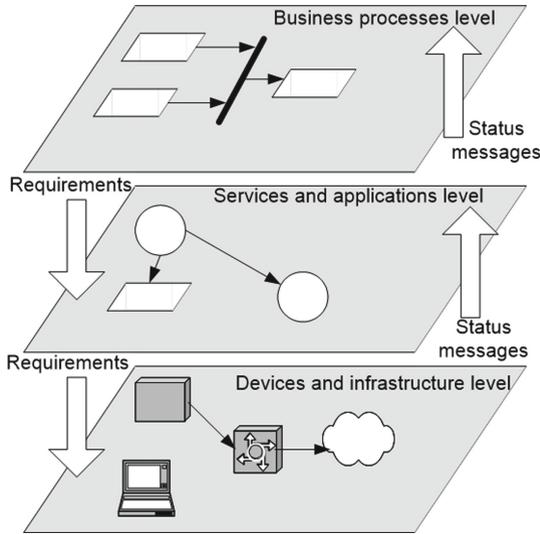


Fig. 2. Conceptual model of the multi-level software modeling system

- Organizational is creating and managing common policies and strategies for testing at the organization level;
- Project management is test planning, monitoring of the state of testing, the definition of criteria for quality, completion of testing;
- Testing is creating test cases and environments, testing performance, defect reporting and executing running test cases.

In this research, we examine the testing processes conducted within a specific project: those belonging to the second and third groups. They do not cover all possible sub-processes and tasks of the software testing. Our goal was to develop some working subset of test tasks and demonstrate the practical feasibility and value of constructing an ontology based on analyzing processes and tasks. To identify all the factors, entities and relationships that influence the performance for each task and process, ontological models were constructed. Figures 3 and 4 show the examples of models for the task of test case execution and the process of testing the fulfillment of the requirement. The tester does test case execution. He uses test-case description, testing environment and test management system to report results. The process of testing requirement fulfillment begins with the preparation of the testing environment and installing all necessary software. The test requirement is defined in the “Specification of Requirements” document. If, as a result of testing, it is found that the requirement is not fulfilled, then the corresponding defect is opened. The test result is reported in the test management system. The analysis of the ontological models of software testing tasks resulted in creating an ontology of software testing (Fig. 5). This ontology was used to implement a software development company’s overnight automatic testing process. The changes made in the code during the working day are integrated

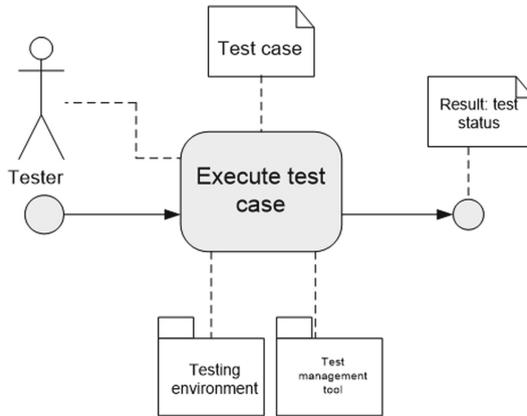


Fig. 3. The ontological model of the task “Test case execution”

into the new version of the product. This version is tested automatically with a minimum set of tests to detect basic functionality disruptions caused by the new code. At the beginning of a new day, developers get a list of defects that need to be addressed. The basic information about testing is stored in an instance of the entity of *AutomatedTesting* in the form of attributes or relations to other entities. This information refers to the entity of the test server and the source of installers and contains a link to the general algorithmic model of automated testing. A generic algorithmic testing model (Fig. 6) is presented as a set of tasks

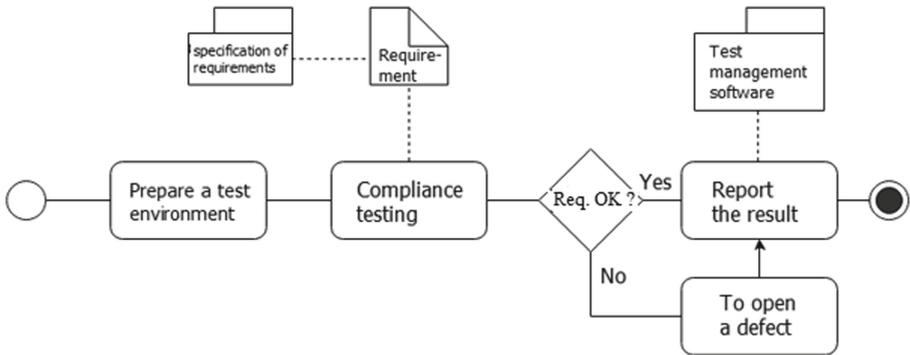


Fig. 4. The ontological model of the task “Compliance testing”

that need to be addressed in automated testing. It contains links to the generic models of the corresponding operations, which reflect the process of solving each task. Such operations, for example, could be *GetInstaller*, *InstallProduct*, *Test*, *UninstallProduct*. In addition to the specification of entities and operations, each model contains the definition of methods for the task execution success checking. For example, after completing the *GetInstaller*, the model assumes that the installer file exists in the specified directory and, if it exists, determines the result of the model execution as successful. Otherwise, the model execution result is determined as unsuccessful with the additional specification of the causes of the error. In addition to operation models, the algorithmic model references the model for determining the following operation and the model for processing the errors. Let's consider the model *GetInstaller*. This model aims to get the installer in the specified directory. This model verifies the availability and copies the installer from the remote server to the local test server using a file copying

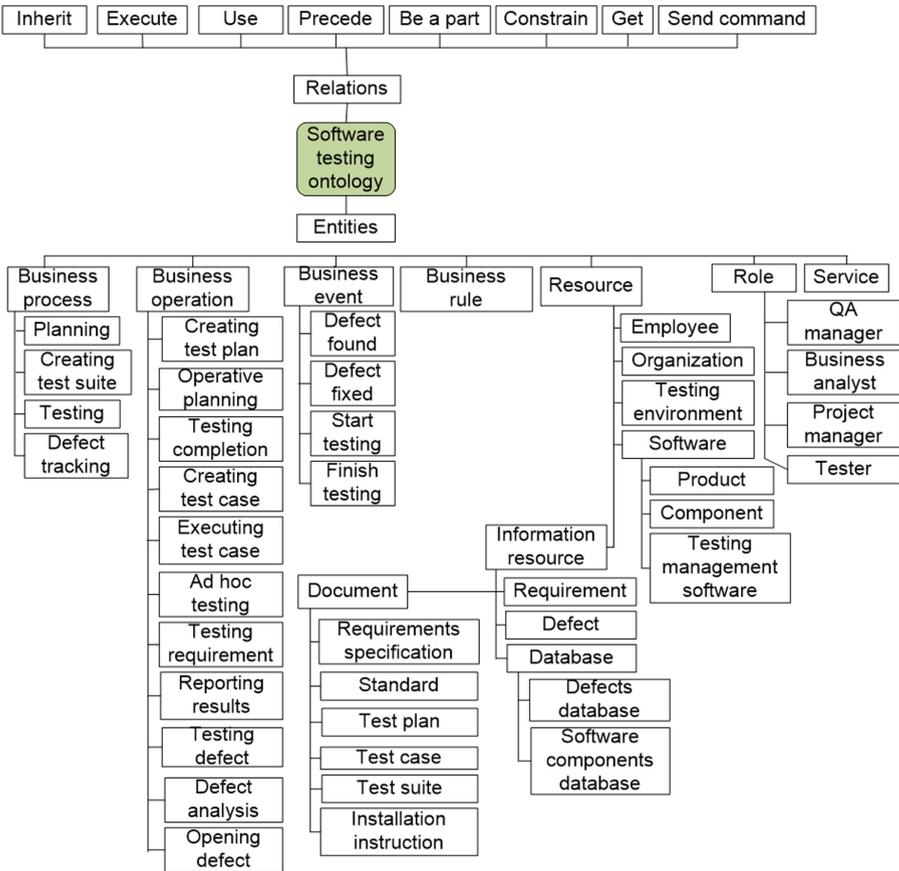


Fig. 5. Entities and relations from software testing ontology

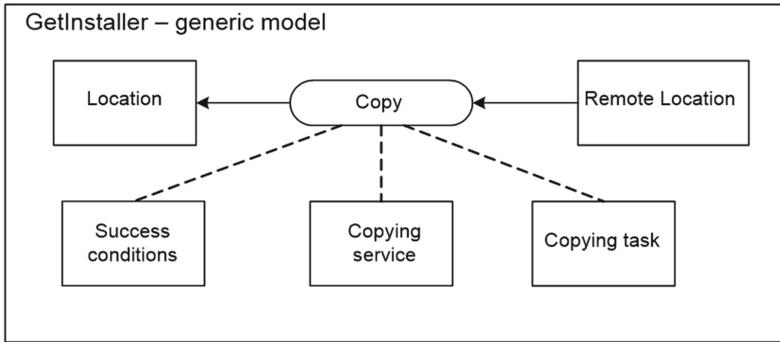


Fig. 6. The generic model *GetInstaller*

service. The generic model of such operation (Fig. 6) contains entities *Remote-Location*, *LocalLocation*, *CopyingService*. The use of such a generic model, which describes the task in its most general form, adds flexibility to the system. Using such a model, the problem can be solved in different ways, simply by specifying different details of the generic model. For example, the *GetInstaller* generic model can be further detailed by using different download services and protocols such as *ftp*, *http*, *from VSS*, or another computer on a local area network. To modify the model, we need only to change the detailed model. Let's compare the generic model *GetInstaller* with a more specific *GetInstallerWithFtp* model (Fig. 7). The entity *RemoteLocation* in the detailed model corresponds to the entity *FtpServer* to the entity *LocationTestServer*. *FtpServer* has attributes describing ftp server's URL, username and password. The entity *CopyingService* in the generic model maps to the entity *FtpService*. Model *GetInstallerWithFtp* specifies additional constraints reflecting the specifics of ftp protocol and service and specific procedures to check whether the file transfer was successful. An automated software testing system was created, tested, and successfully operates using the described approach. *VBScript* scripting language was chosen to implement the system.

HP QuickTest Professional, which also uses *VBScript* as a programming language, is an automated testing tool. The computerized testing system periodically checks for the availability of a new installer file on the developer's site FTP. The installer file is downloaded and unpacked into a specific test directory if available. After that, the installation process is started in automatic mode, which does not require user input. After its completion, the system checks for the success of the installation configures and restarts some system services. The automated testing tool that performs the selected tests is started if the product is successfully installed. The test results are recorded in the protocol, which is sent by e-mail to the developer upon completion of the test. The quality control department employees are also informed by e-mail if the testing process fails. After the test's completion, the tested software product is uninstalled, the test catalog is cleared, and the environment is being prepared for a new test.

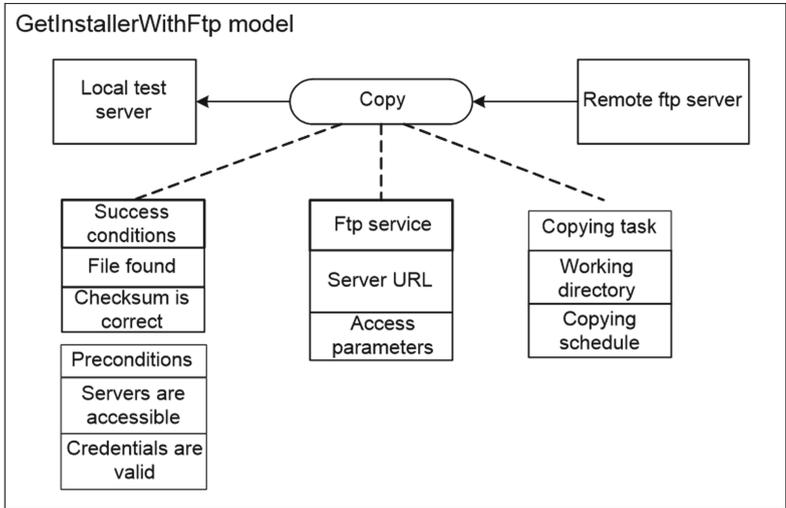


Fig. 7. Structure of model `GetInstallerWithFtp`

The developed system allowed to test systematically several large-sized software products in one night. The operation of the testing system has proven its high reliability, flexibility, and simplicity in modification and development.

4.3 The Modeling Environment for Ontological Models

To study the feasibility and practical usefulness of the model-based software system design approach, a software prototype for the modeling system was developed. This prototype consists of four components, united in a typical application: *Ontology Editor*, *Fact Editor*, *Model Editor* and *Modeler*. Accordingly, the main window of the modeling environment has tabs for ontology, facts, model editors and modeler (Fig. 8). Also are shown the basic constructs used in systems, such as *Attribute types* and *Roles*. Ontological engineer uses the *Ontology Editor* for the creation and modification of ontology. This software component has several functions, such as export and import of ontology; creation, modification and deleting of ontology entities; implementation of inheritance; creation, modification and removal of relations; creation, modification and removal of constraints. Ontologies and other data are stored in XML format. The system supports exporting and importing ontology and all related data sets (facts, models) as separate modeling contexts.

Fact editor is designed to create and modify facts. Every fact is an instance of some entity in the ontology. The user creates facts and also specifies their attribute's values. The system checks the constraints associated with the attributes in the process of fact validation. The system will accept only the fact that was validated successfully. Similarly to facts derived from entities, there are facts derived from relations. Fact-relations additionally contain data about

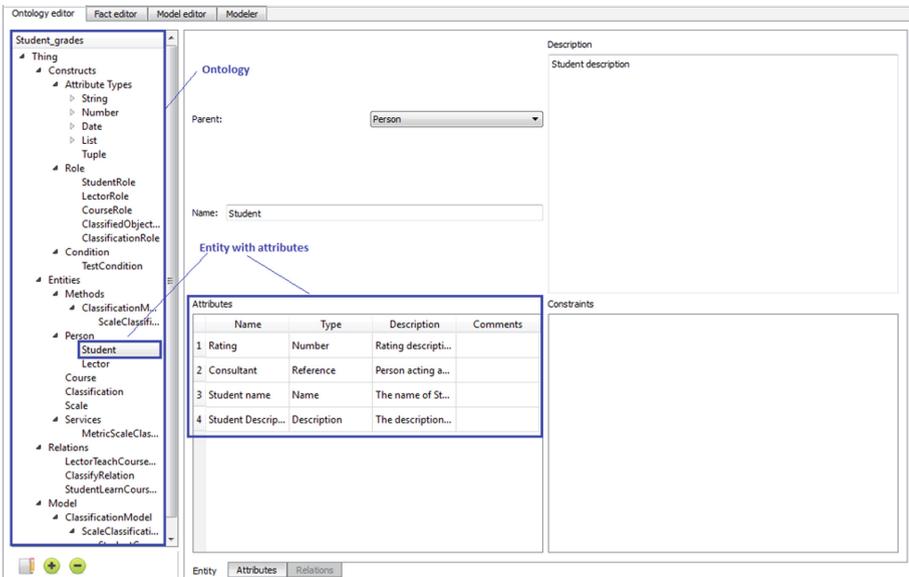


Fig. 8. The main window of modeling environment with ontology editor

roles and facts used to substitute those roles. For example, Fig. 9 is shown the fact-relation *PeterLearnsCourse*, as an instance of relation *StudentLearnsCourse*. Two roles are specified - Student and Course, which are substituted by facts with *Peter* and *ComputerNetworks*. Fact relation has an attribute Rate. *Model Editor* is a central component of the modeling environment. It allows for creating, modifying and deleting models, model metadata and constraints, and operations. Figure 10 shows an example of a simple classification model representation. The roles *ClassifiedObjectRole* and *Classification* are linked by relation *Classify*. In the process of model execution, the fact itself is used, and some objects in this fact's context (for example, the fact's attribute values). Figure 11 shows an example of simple interval classification problem solving using the modeling environment. Developed software prototype modeling environment provides the necessary flexibility for model modification. For example, the change of classification scale requires only attribute update or creating and using another fact of *Scale* type. Likewise, if a classification method is changed, then only the attribute *Method* value of *Classify* relation should be updated. Figure 12 illustrates an example of implementing complex automated testing tasks described previously. *Component Modeler* allows to create of fact-models based on specified models, initializes them by facts from an information base and executes operations in the model. The automated testing model executes the steps of copying the installer from the development server, installing tested software, starting a test script, uninstalling tested software and sending test results back to developers.

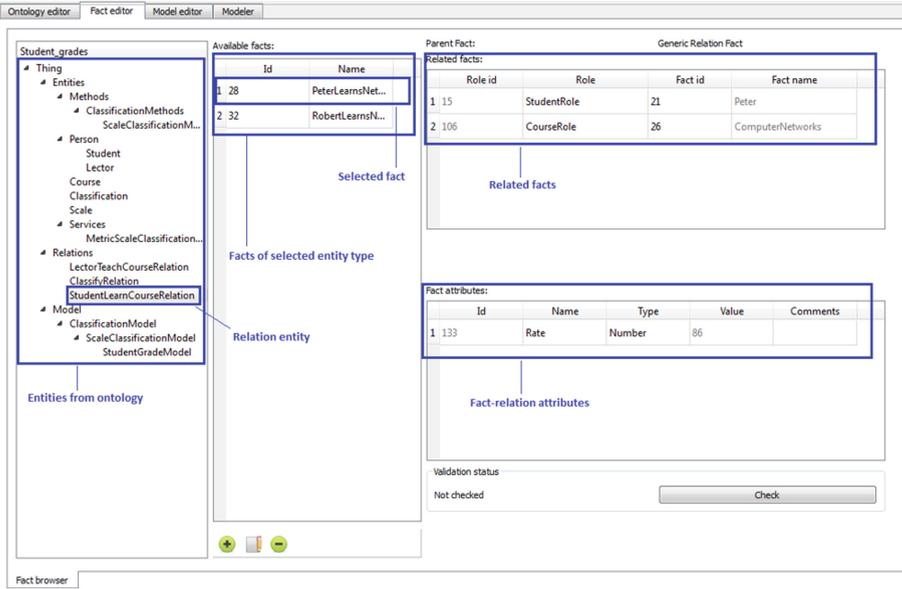


Fig. 9. Fact-relation example

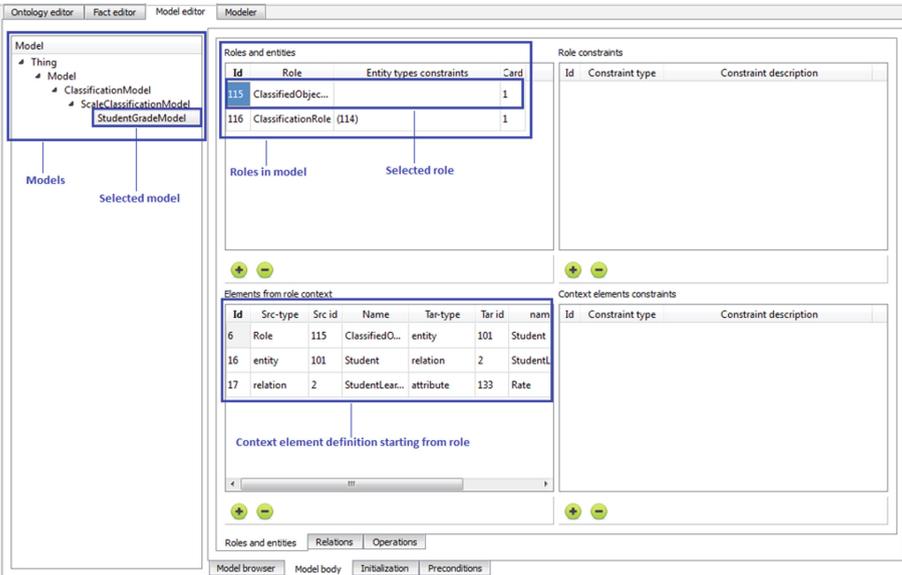


Fig. 10. Model editor

4.4 Analysis of Advantages Using Ontological Models in Software Design

We analyze the advantages of building software based on ontological models compared to traditional software development approaches to assess the degree of software quality improvement at all stages of its life cycle [4, 6–8]. Existing software quality assessment methodologies formulated in standards [1] consider the quality of a software product at various stages of its life cycle through quality models. The underlying quality characteristics are defined in standard ISO/IEC 9126-1. Fifteen software systems of similar complexity (number of functions 7–10, number of entities 7–11) were selected for analysis to obtain numerical estimates of the impact of the use of ontological task models on software quality characteristics. These software systems fall into two groups: web and accounting information systems. Four variants of changes were proposed for each software, and projects were planned to implement these changes. An implementation plan was developed for each project, and the required length of time needed for introducing changes was estimated.

At the conceptualization stage, the software developer solves the problem of constructing the conceptualization of the subject area for the tasks that the software should solve. However, a great deal of work on primary conceptualization and analysis of the domain for different products is duplicated. When using the ontological approach, the developer uses the conceptualization of the domain reflected in the ontology. Avoiding re-conceptualization directly affects quality characteristics such as compliance, accuracy, interoperability, maturity, use of resources at the design stage, portability, and interaction with other products. It is easier to organize the interaction for different software products that share a common conceptualization because they are based on a common set of concepts with the same attributes. Repeated use of a single ontology in a variety of software products achieves a high degree of conceptualization maturity and the stability and reliability that is achieved through repeated and versatile ontology testing during the creation and operation of products based on it. The formula for assessment of the degree of conceptualization reuse is defined as the entities number ratio added to the ontology N_{en-new} to the number of entities existing in ontology before its update N_{en-old} :

$$C_{en-re} = 1 - \frac{N_{en-new}}{N_{en-new} + N_{en-old}} \quad (41)$$

The analysis of the sample projects showed that the reuse rate of concepts for different product pairs varied in the range from 0.1 to 0.88 (Fig. 13). Products based on common conceptualization have better interoperability by avoiding mistakes related to the different domain presentations. The formula for assessing the interoperability deterioration that occurs due to differences in conceptualization for two products A and B. Products based on common conceptualization have better interoperability by avoiding the errors associated with different presentations of the domain. We give an estimate of C_{int} of the deterioration of interoperability that arises from differences in conceptualization for the two

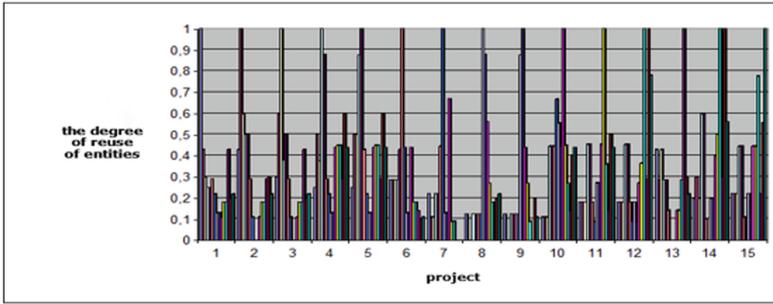


Fig. 13. Evaluation of the degree of reuse of entities

products A and B, in the form:

$$C_{int}(A, B) = \frac{N_{dif}^A + N_{dif}^B}{N_{com}^{AB}} \quad (42)$$

where N_{dif}^A is the number of different elements in the ontology used in software A compared with ontology used in B, N_{dif}^B is the number of different elements in the ontology of software B compared with A, N_{com}^{AB} is the number of common ontology elements reused by A and B.

The use of a single ontology at **the design and coding stage** allows extending the degree of reuse of the code provided by ontological models. When comparing an ontological model-based approach to object-oriented programming, each OOP class has a declarative part represented by declarations of variables and a procedural part represented by a specification of methods. When using ontological models, all models (analogues of methods) have a common declarative part represented by the ontology. When using object-oriented programming, portability is limited by reusing groups of interdependent classes. However, class methods are not portable and can only be used with instances of this class. Using ontologies to build a software system avoids multiple object declarations in different classes, increases code portability, and reduces the use of resources at the coding stage by reusing code. Quality characteristics that improve when using the ontological approach are *portability, and reduced use of resources in the design process*. The following estimates have been proposed to determine the degree of portability of the code:

1. The per cent of software constructs (classes, models, services) in existing code libraries that can be used unchanged:

$$C_{por-code} = \frac{N_{port}}{N_{all}} * 100\%, \quad (43)$$

Note that when using ontological models $C_{por-code} = 100\%$

2. Number of additional constructs (classes, models) that must be included in the product code when using a specific class (model) - $C_{por-add}$. An analysis

of software development projects sample has shown that different software has the same functions that models can represent. The average value of the function reuse rate for the analyzed projects is shown in Fig. 14.

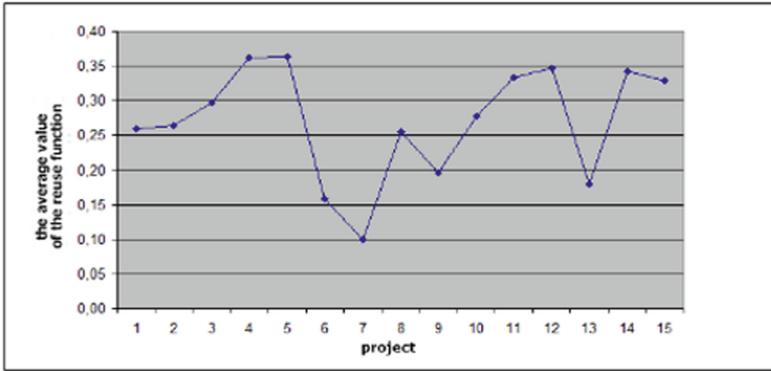


Fig. 14. The average value of the function reuse rate in the analyzed software

At the software testing stage, the use of ontological task models results in reduced testing volumes because only newly developed or modified models are subject to testing. In the traditional approach to software testing, testing is performed based on use-cases. For each use case, a set of test cases is created. If the software has changed, it is difficult to determine the relationship between changes in code and use-cases. Therefore, testers retest all test cases or a significant part of them. Another source of testing process simplification brought by models stems from the fact that a model describes how to solve a particular task which often corresponds to a system’s use case. Therefore, testing use-case means testing the model, reducing the number of test cases and time spent on creating them. Let’s assess the degree of effort reduction for testing in a situation when software is moved to a new (ontological) platform. Assume that both test cases for traditional testing and models for new are available. A metric which reflects the reduction of effort is a ratio between the number of test cases in the new approach N_{mod} to number of test cases N_{old} in traditional one. Also, it takes in consideration the average complexity of each test case $C_{av-tes-old}$ and model $C_{av-tes-mod}$ (measured, for example, as a steps number or testing time estimates):

$$C_{test} = \frac{N_{mod} * C_{av-tes-mod}}{N_{old} * C_{av-tes-old}} \tag{44}$$

An important additional factor in improving the quality of software built using ontologies is that models that are reused in a software product have previously been tested and used in other software. *Testability, stability, and maturity* are key features of the software product quality improved by using ontological models at the testing stage.

At the software maintenance and operation stage, code should be adapted to changes in the business situation or subject area. To evaluate the complexity of software modification traditionally, used the formula:

$$C_{m-old} = T_{ov} \sum_{i=1}^n K_{ov}^i + T_{ch} \sum_{j=1}^m K_{ch}^j \quad (45)$$

where n is the total number of program elements (classes, functions), m is the number of program elements to be modified, K_{ov} is the measure of complexity of the i -th component, T_{ov} is the average time spent on analysis, testing and coding per one program component and one unit of complexity, K_{ch} is the complexity measure of the changing component, T_{ch} is the average time spent extra on analyzing, re-coding, and testing based on one changing software component. Changes in business requirements and subject areas that entail the development of a new version of the software can be of a different nature, making it necessary to analyze them separately. We will consider these types of changes. A. Replacing the method of task execution or forming a new method as a combination of existing ones. The ontology remains unchanged. B. Development of a new method of task execution or refinement of an existing one. C. Updating the domain representation in ontology by introducing new entities, relations, and constraints. Previously existing elements are not updated. D. Updating our understanding of domain resulting in changes in existing ontology. Suppose a software change is only in creating a new task model or modifying an existing one, compared to the traditional approach. In that case, the time spent will mostly be spent developing and testing the new model. However, since the model is relatively simple and no other software components are updated, the complexity of the update, compared to the modification of the traditional way, is insignificant. The complexity of the model update can be estimated using the formula:

$$C_{md} = K_m(T_{an} + T_{mon} + T_{tes}) \quad (46)$$

where C_{md} is the metric of change complexity, K_m is the number of components, T_{an}, T_{mon}, T_{tes} are times spent on analysis, modification and testing per component. The analysis of examples of changes in the method of solving the problem for the sample of real projects demonstrated a significant (5–8 times) reduction of time for making the change when using task models. Analysis of the duration of modifications, in this case, showed significant advantages of the model approach compared to the traditional one (Fig. 15). If you need to develop a new model or modify an existing one, the time spent on creating and testing the model increases compared to the case A. Analysis of changes from the sample of real projects showed that the reduction of time, in this case, reaches 2–5 times in comparison with the traditional approach (Fig. 16). In the case of the introduction of new entities, relationships or constraints in ontology, while preserving its existing parts without modification, the ontology designer validates the proposed changes. It analyses these changes for the absence of conflicts with existing ontology elements and justifies the reasons for modifying the ontology.

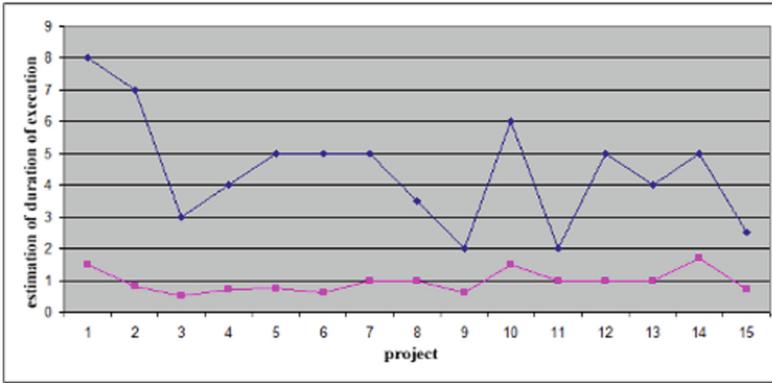


Fig. 15. Comparison of estimates of the duration of the type A change in software systems

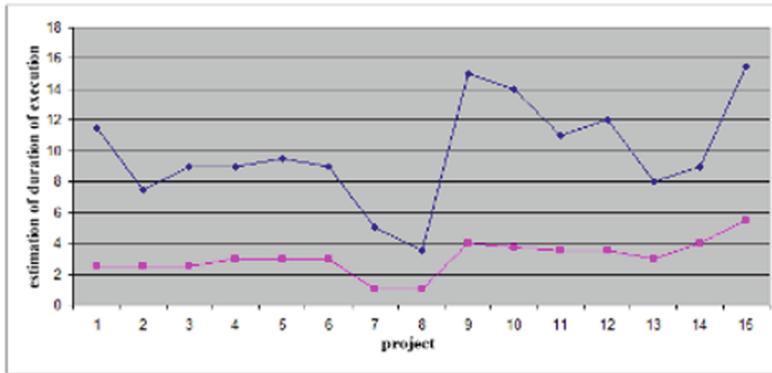


Fig. 16. Comparison of estimates of the duration of the type B change in software systems

The ontology modification is carried out in the ontology editor. The complexity of the modification depends on the number of new elements added to the ontology:

$$C_{md-ont-add} = K_{mod}(T_{an} + T_{mon}) + T_{md} \tag{47}$$

where $C_{md-ont-add}$ is complexity metric, K_{mod} is the number of components added to ontology, T_{an} is time spent on analysis and validation, T_{mon} is time spent on modification of ontology for one component, T_{md} is time spent on modification of dependent components of ontology. Analysis of changes from the sample of real projects showed that the reduction of time, in this case, reaches 1.5–3 times compared to the traditional approach (Fig. 17). The change

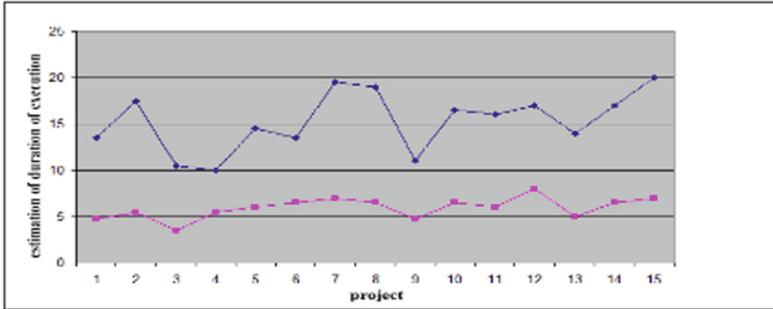


Fig. 17. Comparison of estimates of the duration of the type C change in software systems

in existing elements of ontology is the change in understanding of the domain and the discovery of errors in the ontology. Changing existing ontology elements requires changing and retesting all models that work with modified elements. This task is quite labor-intensive since the number of models in the model repository is significant. However, the architecture and functional capabilities of the modeling system simplify the finding of all models influenced by changes in specific ontology components. For each updated model, it is necessary to analyze the effects of changes, determine the scope of the necessary modifications, update the model and retest it:

$$C_{md-ont-rev} = (K_{on} * T_{on}) + K_{md-on}(T_{an} + T_{md-mod} + T_{tst}) \quad (48)$$

where $C_{md-ont-rev}$ - is metric of complexity, K_{on} - is the number of changed ontology components, T_{on} - time spent on modification of one ontology component, K_{md-on} the number of models updated, T_{an} - is time spent on analysis and design of changes in the model, T_{md-mod} - is time spent on the model update, T_{tst} - is time spent on model testing. The analysis of changes implementation in case D for both traditional and ontological models based on approaches has shown that software modification time has substantially increased. However, the modification times for the model based on the process are still (1,2–2 times) shorter (Fig. 18). The software update and testing tasks are carried out in the modeling

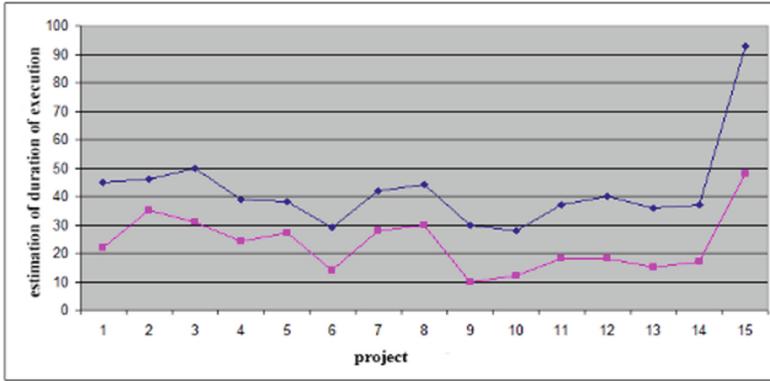


Fig. 18. Comparison of estimates of the duration of the type D change in software systems

environment and do not take much time. The greatest amount of time is spent analyzing and designing changes in models. When comparing the complexity of software modification in cases A-D, it is impossible to give accurate estimates of complexity without considering the specifics of projects and development teams. At the same time, we can rank metrics based on expert assessments. Thus, we get an ordered set of metrics with the growing complexity:

$$(C_{md}, C_{md-ont-add}, C_{md-ont-rev}, C_{m-old}), \tag{49}$$

The use of ontological models for software construction leads to the improvement of such quality characteristics as correctness, number of defects, compliance with functional requirements and ease of implementation of changes. The further development of the modeling system involves expanding the functions of interaction of models, implementing mechanisms of logical inference based on ontology, and automatic and asynchronous execution of models.

5 Conclusions

The use of ontological models increases the efficiency of solving problems in different subject areas. In particular, [6–8] demonstrated this approach’s effectiveness in solving decision support tasks, controlling access to information resources, automated software testing, and customer service. Theoretical and experimental studies on the development of methods of using ontological models in business intelligence systems have been carried out at Information Systems and Networks Department of Lviv Polytechnic National University for several years. In particular, an experimental model of a software package for modeling systems using an ontological approach and models was developed. The mathematical model of knowledge representation presented in this paper serves as a formal basis for constructing the instrumental complex and validating it. As a

result of the research, the actual scientific and applied problem of application of ontological models of problems for the construction of intelligent software systems was solved. The expediency of using ontological models of problems was substantiated, and the mathematical formalization of knowledge representation was developed. The software system functioning structure and principles based on ontological models were determined. The advantages of using ontological models compared to the traditional approach were analyzed. The prototype of the instrumental modeling system was developed. Practical implementation and verification were carried out at the leading companies in the Ukrainian IT industry, such as Softserve, Eleks, and the received professional feedback is commendable. At the same time, the question of using descriptive logic and inference mechanisms in the system based on problem models remained unresolved in the context of the obtained results. The methods of interaction of models, implementation of multivariate calculations, and context-dependent model-based calculations require further development.

References

1. 29119-5-2016 - ISO/IEC/IEEE international standard, software and systems engineering – software testing. <https://standards.ieee.org/standard/29119-5-2016.html>
2. Task definition. <http://www.businessdictionary.com/definition/task.html>
3. Babichev, S., Lytvynenko, V., Skvor, J., Korobchynskyi, M., Voronenko, M.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction, vol. 8478452, pp. 336–341 (2018)
4. Bakurova, A., Pasichnyk, M., Tereschenko, E.: Development of a productive credit decision-making system based on the ontology model, vol. 2870, pp. 580–589 (2021)
5. Balmelli, L., Brown, D., Cantor, M., Mott, M.: Model-driven systems development. *IBM Syst. J.* **45**, 569–585 (2006). <https://doi.org/10.1147/sj.453.0569>
6. Burov, Y.: Business process modelling using ontological task models. *Econtechmod* **1**, 11–23 (2014)
7. Burov, Y., Pasichnyk, V.: Software systems based on ontological task models. LAP LAMBERT Academic Publishing (2018)
8. Burov, Y., Vysotska, V., Kravets, P.: Ontological approach to plot analysis and modeling, vol. 2362, pp. 22–31 (2019)
9. Evans, E.: *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, Boston (2003)
10. Gruber, T.: A translation approach to portable ontology specifications. *Knowl. Acquis.* **5**, 199–221 (1993). <https://doi.org/10.1006/knac.1993.1008>
11. Ikeda, M., Seta, K., Kakusho, O., Mizoguchi, R.: An ontology for building a conceptual problem solving model, pp. 126–133 (1998)
12. Johnson, P., Johnson, H., Waddington, R., Shouls, A.: Task-related knowledge structures: analysis, modelling and application, pp. 35–62 (1988)
13. Kaur, U., Singh, G.: A review on software maintenance issues and how to reduce maintenance efforts. *Int. J. Comput. Appl.* 6–11 (2015). <https://doi.org/10.5120/ijca2016912414>
14. Koo, B., Simmons, W., Crawley, E.: Algebra of systems: a metalanguage for model synthesis and evaluation. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **39**(3), 501–513 (2009). <https://doi.org/10.1109/TSMCA.2009.2014546>

15. Koskinen, J., Lahtonen, H., Tilus, T.: Software maintenance costs estimation and modernisation support. Information Technology Research Institute. https://static.aminer.org/pdf/PDF/000/364/724/estimating_the_costs_of_software_maintenance_tasks.pdf
16. Lehman, M., Ramil, J.: Software evolution-background, theory, practice. *Inf. Process. Lett.* **88**, 33–44 (2003). [https://doi.org/10.1016/S0020-0190\(03\)00382-X](https://doi.org/10.1016/S0020-0190(03)00382-X)
17. Litvinenko, V., Burgher, J., Vyshemirskij, V., Sokolova, N.: Application of genetic algorithm for optimization gasoline fractions blending compounding, vol. 1048134, pp. 391–394 (2002)
18. Lytvynenko, V., Lurie, I., Krejci, J., Savina, N., Taif, M.: Two step density-based object-inductive clustering algorithm, vol. 2386, pp. 117–135 (2019)
19. Lytvynenko, V., Savina, N., Krejci, J., Yakobchuk, M., Kryvoruchko, O.: Bayesian networks' development based on noisy-max nodes for modeling investment processes in transport, vol. 2386 (2019)
20. Osterwalder, A.: The business model ontology a proposition in a design science approach. Ph.D. thesis, Université de Lausanne, Faculté des hautes études commerciales (2004). <http://www.hec.unil.ch/aosterwa/PhD/Osterwalder-PhD-BM-Ontology.pdf>
21. Pezzulo, G., Calvi, G.: Schema-based design and the AKIRA schema language: an overview. In: Butz, M.V., Sigaud, O., Pezzulo, G., Baldassarre, G. (eds.) *ABiALS 2006. LNCS (LNAI)*, vol. 4520, pp. 128–152. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74262-3_8
22. Raubal, M., Kuhn, W.: Ontology-based task simulation. *Spatial Cogn. Comput.* 15–37 (2004). https://doi.org/10.1207/s15427633scc0401_3
23. Ross, R.: *Principles of the Business Rule Approach*. Addison-Wesley Professional, Boston (2003)
24. Schröder, F.: *Ontology engineering for robotics* (2017). <https://vixra.org/pdf/1711.0360v1.pdf>
25. Van Welie, M.: An ontology for task world models. In: Markopoulos, P., Johnson, P. (eds.) *Design, Specification and Verification of Interactive Systems*, pp. 57–70 (1998). https://doi.org/10.1007/978-3-7091-3693-5_5



Neural Network Analysis of Evacuation Flows According to Video Surveillance Cameras

Oleksandr Khlevnoi , Nazarii Burak^(✉) , Yurii Borzov , and Diana Raita 

Lviv State University of Life Safety, Lviv, Ukraine
nazar.burak.ac@gmail.com

Abstract. The paper is dedicated to establishing the relationship between the speed of evacuation participants and their density on the basis of neural network analysis of their movement using video surveillance systems. An analytical review of modern methods for studying the evacuation flows movement parameters was carried out. Despite the availability of modern software-simulating complexes, the movement parameters of the evacuation participants for these complexes are determined by establishing relationships between the flow density and the speed of the evacuation participants. These dependencies can be identified mainly by the results of field observations and processing of video recordings of these observations, which takes a lot of time and requires a lot of effort and indicates the need for automation and optimization of the video processing process. The tools for analyzing and classifying images, as well as detecting and classifying moving objects in a video stream deserve special attention, as they can be successfully used to study the problems of evacuation in case of fire. It was concluded that the tools of the OpenCV library are the most convenient in use. This library was applied for recognizing evacuation participants of various mobility groups. Apart from this the convolutional neural network was used for processing movement parameters. The neural network demonstrated best accuracy (81%) in the closest to the camera part of the evacuation route up to 5 m at a human flow density of up to 3 person per square meter.

Keywords: Artificial neural networks · Computer vision · Machine learning · Emergency evacuation · Evacuation speed · Flow density

1 Introduction

At the stage of construction and operation of various facilities, it is necessary to ensure compliance with the level of fire safety of people and the individual fire risk values to the established standards [4]. The evacuation duration from buildings and structures must be taken into account when calculating these indicators [12].

The analysis of the current methods used for calculating the evacuation duration showed that the speed and intensity of movement as well as the density of human flows are the determining parameters of evacuation process. The relationship between these parameters can be described as follows:

$$V_D = V_0 \left[1 - a \ln \frac{D}{D_O} \right], \text{ at } D > D_O \quad (1)$$

where V_D is the evacuation flow speed at flow density D , meter per second (m/s); D - the current value of the flow density, person per square meter (per/m^2); D_O - threshold value of flow density, after which the density becomes a factor influencing the evacuation flow speed, person per square meter (per/m^2); V_0 - the value of the free movement speed (when the human flow density doesn't affect the speed), which depends on the type of path and emotional state of people, meters per minute (m/min); a - coefficient that reflects the degree of influence of human flow density on its speed.

Modern software and modeling systems that allow calculating the time of evacuation from buildings and structures use such dependencies as initial data. That is, in order to ensure high accuracy of the program, it is first necessary to conduct an experimental study of the evacuation parameters on real objects.

In the paper, according to the analysis of literature, a block diagram for optimization process of forming evacuation parameters empirical databases was proposed.

Due to the proposed block diagram, using the results of research conducted by the authors, a training sample was formed and the convolutional neural network was trained.

For certain CCTV recordings a model to convert the obtained speed values into the specified units were developed.

The use of the neural network showed satisfactory results in the observation area, located at a distance of up to 5 m from the video camera and at a human flow density of up to 4 person per square meter.

The results of the work can be used to develop intelligent evacuation warning and control systems, interactive evacuation plans and to conduct research on the parameters of mixed evacuation flows.

2 Problem Statement

A significant problem is that such dependencies can differ significantly for different sections of roads (horizontal sections, stairs, ramps, etc.), as well as for different composition of the flow (e.g. for evacuation flow consisting of children values V_0 , a and D_o will differ from similar values for flow consisting of adults).

In addition, the development of inclusive education and the implementation of the principle of accessibility mean that people with special needs are free to visit public facilities. This affects the change in the structure of human flows on such objects and requires intensification of the study of the mixed flows movement parameters [19].

Despite the fact that the development of modern software and modelling systems and intelligent evacuation management systems significantly improves the calculation accuracy of evacuation duration, the problem of establishing relationships between speed and density of human flows is still solved by complex and time-consuming processing the results of field observations.

Therefore, solving the problem of optimization and automation of this process is an extremely important task.

The authors of this research consider that the implementation and application of neural networks, which have become an integral part of society today, can solve this problem.

3 Literature Review

A large number of works are devoted to the study of evacuation from various buildings. Research [13] is devoted to evacuation from health care facilities and institutions for the elderly people. In [21] research was conducted in preschool education institutions, and in [20, 25] - in secondary school education institutions. Much attention is also paid to the evacuation of people in wheelchairs [23, 24, 27] and evacuation of children with special needs [29]. In these works, the evacuation flow movement parameters were determined based on the processing of video camera recordings without the use of video stream analysis software tools.

At the same time, based on the analysis of a number of studies [7, 9, 11, 17, 26, 30], it can be concluded that the tools for analyzing and classifying images, as well as detecting and classifying moving objects in a video stream, make it possible to obtain high-quality results. Studies related to the determination of the parameters of pedestrians during traffic [7] deserve special attention, as they can be successfully used to study the problems of evacuation in case of fire.

It should be noted that there are a large number of implementations of neural networks in the public domain [3, 8, 15, 18, 22]. These neural networks can be successfully used for the recognition and analysis of moving objects. This opens up good prospects for obtaining large amounts of data on the evacuation flows movement and establishing relationships between the speed of movement and the density of people in these flows.

The purpose of the paper is to increase the speed and accuracy of the formation of empirical databases of evacuation flow parameters based on the analysis of video stream data.

4 Stepwise Procedure of the Evacuation's Parameters Determining Optimisation

The processing of the results in the above evacuation studies occurred in the following sequence:

- a frame from each camera was transferred to a random graphics editor, in which a grid was applied to the image. This grid divided the entire observation area into squares of 1×1 m (or larger);

- the grid (without the frame) was saved in *PNG* format;
- the grid was superimposed on the corresponding video using an arbitrary application for video editing;
- the grid was used for determination both the distance passed by the flow participant for the corresponding period of time (and therefore - the speed), and the number of people in a certain section of the area (the flow density) (Fig. 1)



Fig. 1. Examples of grids for processing video recordings [28]

Speed values obtained due to the results of field observations were grouped by density and variation interval arrays were formed in the order of the density growth. Based on the obtained values, the regression dependencies of the movement speed on the density were determined. The results were checked by the way of the CCTV camera records analysis on real objects. It is possible to increase the speed and accuracy of the empirical data obtaining process through neural network analysis of evacuation flows.

The optimization scheme for this process is shown in Fig. 2.

The task of recognizing evacuation participants on video can be represented by a sequence of several stages:

- at the first stage, it is necessary to split the video stream into frames and prepare the frames for object recognition;
- the second stage involves the process of recognizing the participants in the evacuation in each frame by the means of neural network;
- at the third stage, the evacuation participants identified in the each frame are compared with the evacuation participants detected in previous frames;
- the fourth stage - highlighting the detected objects on the video.

As already noted, there are ready-made software libraries that allow recognizing objects in a video stream. Based on the analysis of these resources, we can conclude that the tools of the OpenCV library are the most convenient in

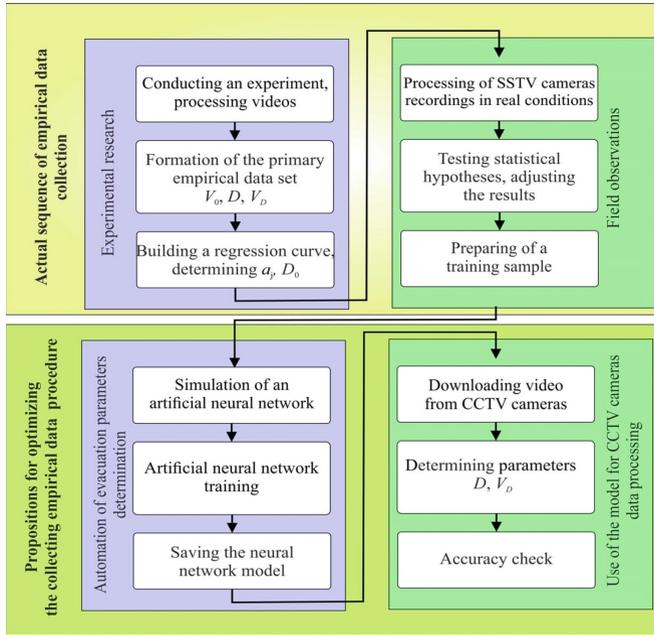


Fig. 2. General block diagram for optimization of the evacuation parameters determining process

use [2]. This library can be applied for recognizing evacuation participants of various mobility groups (including those who use wheelchairs and those who use crutches supports). Apart from this the convolutional neural network has to be used for processing movement parameters.

5 Experiment, Results and Discussion

To process the video stream, the architecture of the convolutional neural network proposed in [5] was chosen. The architecture is freely available.

The initial data set for the neural network were frames processed using OpenCV. Since the video stream was obtained using a stationary video surveillance camera, frame masking was applied to the sections where evacuation participants were not detected (Fig. 3).

To train the neural network, the results of the experiments conducted at the Lviv State University of Life Safety [19] were used. The sample consisted of videos, the total number of participants on the frames was more than 35,000 people. The training duration was 400 epochs.

In the training set, all participants in the evacuation were divided into 2 categories: participants moving independently ($M1$) and uncategorized (atypical) participants ($M5$), which include participants using wheelchairs, participants using crutches etc.).

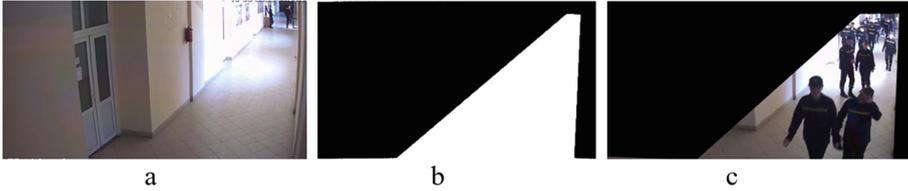


Fig. 3. Image masking: a) the frame of a CCTV camera; b) the mask used in OpenCV; c) applying the mask to the frame

The network was trained using tensor processors on the cloud platform for machine learning Google Colaboratory [1]. All the detected participants were bounded with a rectangular box with the coordinates of the geometric center (\dot{x}, \dot{y}) .

The movement of the object in the interval between two adjacent frames was determined using the SORT (Simple Online and Real-time Tracking) algorithm [6]. This algorithm provides the ability to calculate the participant movement speed (in pixels per unit of time). The specified characteristics of each box (the ratio between the height and width and the area) in each frame are compared with the characteristics of the boxes in the previous frame. The use of the SORT algorithm makes it possible to identify boxes that existed in previous frames and to determine their movement, as well as identify evacuation participants who appeared in the frame for the first time.

After that it was necessary to convert the obtained values in pixels per second to values in meters per minute. To implement these calculations, a horizontal section of the evacuation route measuring 22×2.4 m was chosen. This section was located in the frame of a video surveillance camera.

It should be noted that the distance traveled in pixels in the section of the evacuation route that is more distant from the camera is significantly greater in meters than the same distance in pixels in the section located near the surveillance camera. To avoid conversion errors the evacuation area in the frame was divided into 4 equal parts (I–IV) 5.5 m long each [14].

Split points 1–10 were plotted on a CCTV camera frame. This made it possible to compare the coordinates of these points on the image and the real object (Fig. 4).

The model presented in [10] was used. According to this model, the position of each evacuation participant in real coordinates can be determined as follows:

$$x = \frac{k_1\dot{x} + k_2\dot{y} + k_3}{k_7\dot{x} + k_8\dot{y} + 1}, \tag{2}$$

$$y = \frac{k_4\dot{x} + k_5\dot{y} + k_6}{k_7\dot{x} + k_8\dot{y} + 1}, \tag{3}$$

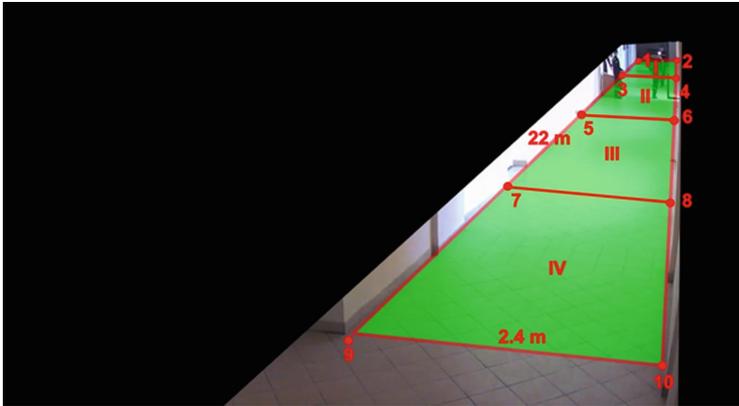


Fig. 4. The section of the evacuation route in the frame of the CCTV camera and its actual size

where (x, y) are the real coordinates of the evacuation participant; (\dot{x}, \dot{y}) - coordinates of the box center in the frame; $k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8$ - conversion coefficients.

From Eqs. (2) and (3) for each of the points 1–10 the system of two equations was written:

$$k_1\dot{x} + k_2\dot{y} + k_3 - k_7\dot{x}x - k_8\dot{y}x - x = 0 \tag{4}$$

$$k_4\dot{x} + k_5\dot{y} + k_6 - k_7\dot{x}y - k_8\dot{y}y - y = 0 \tag{5}$$

Thus, we obtain 10 systems of equations. Substituting the known values for each of the 10 points, we can solve the systems of equations and find the values of the transformation coefficients. Using the obtained coefficients it is possible to calculate the movement of the evacuation participant between two adjacent frames of the video stream.

Each evacuation participant after identification gets a certain number and spends a certain amount of time on the evacuation area (field of view of the camera). This time, consequently, can be written as a certain number of frames n . Immediately after the object is identified, the distance it moves between two adjacent frames can be calculated:

$$l_{n+1} = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \tag{6}$$

If the evacuation participant is found for the first time, his initial movement is accepted as $l_1 = 0$.

The instantaneous speed of the evacuation participant is determined by:

$$v_{n+1} = l_{n+1}w, \tag{7}$$

where w is the camera frequency, frames per second (*fr./s*).

This value is not convenient to use, because in the process of walking a person’s speed changes abruptly and differs in each frame. To get the smoothed speed value the Kalman filter [16] was used:

$$V_{n+1} = K \cdot v_{n+1} + (1 - K) \cdot v_n, \tag{8}$$

where K is the Kalman transfer coefficient.

Since, after processing the video stream by the neural network, all data about the detected evacuation participants were recorded in a separate file, it was important to achieve better visualization. For this purpose a special application was developed. After processing the video file by the neural network the application allows submitting a similar video file, in which all the evacuation participants are highlighted with frames with the calculated speed values above (Fig. 5).

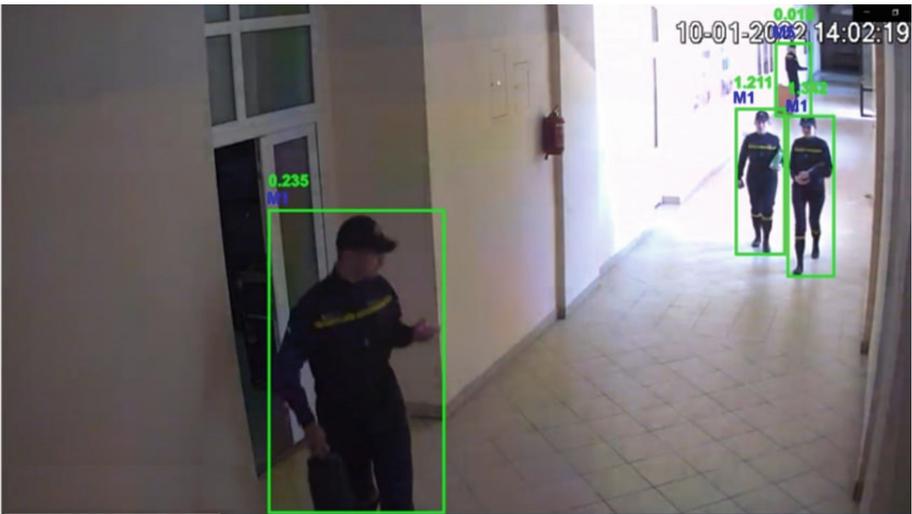


Fig. 5. The result of processing the video stream by the neural network

To verify the accuracy of speed determination, other results of the already mentioned studies [19] were used, where the movement parameters were determined personally by the authors. Figure 6 presents the comparative results of determining the evacuation speed depending on the human flow density.

Evacuation speed values were grouped into variation series by density. An average value was determined for each density interval. This made it possible to compare the results obtained using the neural network with the above results (Table 1).

Figure 7 presents the comparative results of determining the evacuation speed depending on the human flow density.

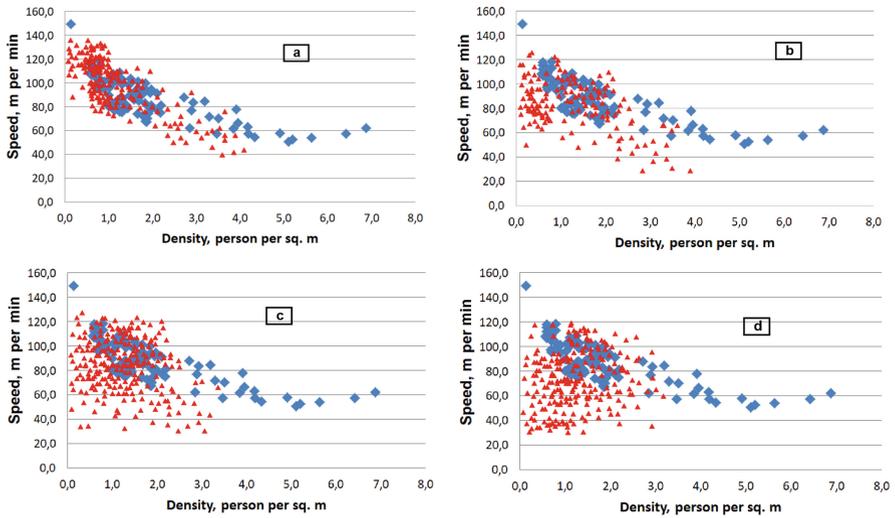


Fig. 6. Evacuation speed and flow density, determined by the neural network (red) and during the experiment (blue): a) section IV; b) section III; c) section II; d) section I

Table 1. Experimental study of the evacuation speed

Density, <i>person per m²</i>	Speed obtained without CNN	Results, obtained with CNN							
		Section IV		Section III		Section II		Section I	
		Speed, <i>m/min</i>	Acc., %	Speed, <i>m/min</i>	Acc., %	Speed, <i>m/min</i>	Acc., %	Speed, <i>m/min</i>	Acc., %
0-1	111	106	95,28	82	73,71	80	71,91	61	54,83
1-2	88	92	95,41	83	94,36	82	93,23	66	69,18
2-3	77	62	80,38	47	60,94	50	64,83	63	78,37
3-4	70	57	81,43	45	64,29	–	–	–	–

Analysis of the results shows that the neural network demonstrated best accuracy (more than 80%) in the closest to the camera part of the evacuation route (IV) at a human flow density of up to 4 person per square meter.

At higher values of the flow density, the participants are placed very tightly, due to which the quality of recognition deteriorates. As the distance to the camera increases, the dispersion of the results increases and the accuracy of speed determination decreases to 54% in section I. It should also be noted that in sections I and II the neural network is not able to correctly identify evacuation participants at a density of more than 3 person per square meter.

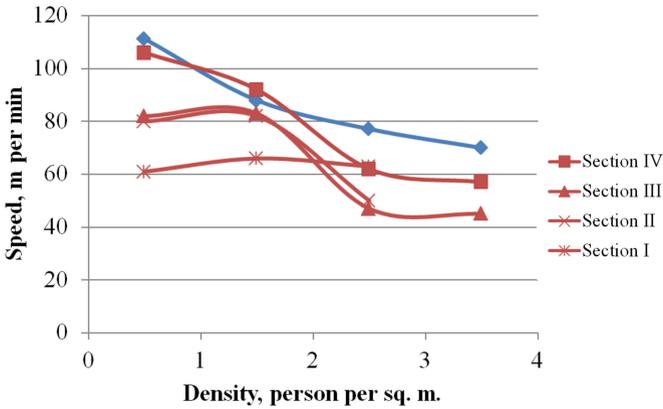


Fig. 7. Dependence of evacuation speed on flow density, determined by the neural network (red), and during the experiment (blue)

6 Conclusions

The conducted research deals with neural network analysis of evacuation flows according to video surveillance cameras.

Methods and means for calculating the evacuation time in case of fire are constantly evolving. This is especially true for software-simulating complexes. Modern technologies always open up new possibilities. At the same time, the acquisition of initial data for calculation in most cases is still carried out by researchers through the processing of video recordings and takes a lot of time. It should be noted a large number of implementations of neural networks. These networks can be successfully used for the recognition and analysis of moving objects, giving good prospects for obtaining large amounts of data on the evacuation flow movement and establishing relationships between the speed of movement and the density of people in these flows.

Authors of the research proposed optimization scheme of the evacuation parameters determining process. A convolutional neural network was used to determine the speed of evacuation participants. Simple Online and Real-time Tracking algorithm was used to calculate the participant movement speed.

The results, obtained using a convolutional neural network, were compared with similar data determined directly by the authors of the paper. For human density up to 4 person per square meter at a distance of up to 5 m from the camera more than 80% accuracy was achieved. Determining the parameters of human flow by means of a convolutional neural network can significantly accelerate experimental research and the formation of an empirical evacuation parameters database.

Further research should be aimed at improving the accuracy of the neural network at higher values of human flow density, as well as to determine the parameters of movement of people with limited mobility.

References

1. Google colabatory. <https://colab.research.google.com>
2. Open source computer vision librar. <https://github.com/opencv/opencv>
3. Yolov5 wheelchair detector. <https://newjerseystyle.github.io/en/2021/YoloV5-Wheelchair-detector/>
4. DSTU 8828: 2019. Fire safety. Terms (2018). http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=82138
5. Abdulla, W.: Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow (2017). https://github.com/matterport/Mask_RCNN
6. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468 (2016). <https://doi.org/10.1109/ICIP.2016.7533003>
7. Cai, Y., et al.: Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video. *IEEE Trans. Intell. Transp. Syst.*, 1–16 (2021). <https://doi.org/10.1109/TITS.2021.3052908>
8. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: realtime multi-person 2d pose estimation using part affinity fields. *CoRR* **abs/1812.08008** (2018). <http://arxiv.org/abs/1812.08008>
9. Chen, C.Y., Liang, Y.M., Chen, S.W.: Vehicle classification and counting system. In: ICALIP 2014 - 2014 International Conference on Audio, Language and Image Processing, Proceedings, pp. 485–490, January 2015. <https://doi.org/10.1109/ICALIP.2014.7009841>
10. Grents, A., Varkentin, V., Goryaev, N.: Determining vehicle speed based on video using convolutional neural network. *Transp. Res. Procedia* **50**, 192–200 (2020). <https://doi.org/10.1016/j.trpro.2020.10.024>
11. Huang, Y.Q., Zheng, J.C., Sun, S.D., Yang, C.F., Liu, J.: Optimized yolov3 algorithm and its application in traffic flow detections. *Appl. Sci.* **10**(9) (2020). <https://doi.org/10.3390/app10093079>. <https://www.mdpi.com/2076-3417/10/9/3079>
12. Hulida, E., Pasnak, I., Koval, O., Tryhuba, A.: Determination of the critical time of fire in the building and ensure successful evacuation of people. *Periodica Polytechnica Civil Eng.* **63**(1), 308–316 (2019). <https://doi.org/10.3311/PPci.12760>. <https://pp.bme.hu/ci/article/view/12760>
13. Istratov, R.: Rationing of fire safety requirements for evacuation routes and exits in hospitals of social institutions for servicing elderly citizens. *Dis. Cand. Tech. Sci. Acad. SES Russia* (2014)
14. Ivanov, Y., et al.: Adaptive moving object segmentation algorithms in cluttered environments, pp. 97–99, February 2015. <https://doi.org/10.1109/CADSM.2015.7230806>
15. Joo, H., et al.: Panoptic studio: a massively multiview system for social motion capture. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3334–3342 (2015). <https://doi.org/10.1109/ICCV.2015.381>
16. Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(1), 35–45 (1960). <https://doi.org/10.1115/1.3662552>
17. Ke, R., Kim, S., Li, Z., Wang, Y.: Motion-vector clustering for traffic speed detection from UAV video. In: 2015 IEEE First International Smart Cities Conference (ISC2), pp. 1–5 (2015). <https://doi.org/10.1109/ISC2.2015.7366230>
18. Khelvas, A., Gilya-Zetinov, A., Konyagin, E., Demyanova, D., Sorokin, P., Khafizov, R.: Improved 2D human pose tracking using optical flow analysis. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) *IntelliSys 2020. AISC*, vol. 1251, pp. 10–22. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-55187-2_2

19. Khlevnoi, O.: Standardization of fire safety requirements for evacuation routes and exits in secondary education institutions with inclusive education. Ph.D. thesis, Lviv State University of Life Safety (2021)
20. Khlevnoi, O., Kharyshyn, D., Nazarovets, O.: Problem issues of evacuation time calculation during fires in preschool and secondary education institutions with inclusive groups. *Fire Saf.* **37**, 72–76 (2021). <https://doi.org/10.32447/20786662.37.2020.11>
21. Kholshchevnikov, V., Samoshin, D., Parfyonenko, A., Belosokhov, I.: Study of children evacuation from pre-school education institutions. *Fire Mater.* **36**, 349–366 (2012). <https://doi.org/10.1002/fam.2152>
22. Kocabas, M., Karagoz, S., Akbas, E.: MultiPoseNet: fast multi-person pose estimation using pose residual network. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV 2018*. LNCS, vol. 11215, pp. 437–453. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_26
23. Kovalyshyn, V., Khlevnoi, O., Kharyshyn, D.: Primary school-aged children evacuation from secondary education institutions with inclusive classes. *Sci. Eur.* **1**(60), 53–56 (2020). <https://doi.org/10.24412/3162-2364-2020-60-1-53-56>
24. Miyazaki, K., et al.: Behaviour of pedestrian group overtaking wheelchair user. In: *Pedestrian and Evacuation Dynamics: Proceedings of the 2nd International Conference*, pp. 267–278, January 2003
25. Nizhnyk, V., Teslenko, O., Tsybalytsky, S., Kravchenko, N.: Regarding the calculation of the time of evacuation of children from schools and preschools in case of fire. *Sci. Bull. Civil Prot. Fire Saf.* **1**, 81–87 (2016)
26. Peleshko, D., Rak, T., Peleshko, M., Izonin, I., Batyuk, D.: Two-frames image super resolution based on the aggregate divergence matrix. In: *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, pp. 235–238 (2016). <https://doi.org/10.1109/DSMP.2016.7583548>
27. Rubadiri, L.: Evacuation modelling of mixed-ability populations in fire emergencies. Ph.D. thesis, University of Central Lancashire (1994)
28. Samoshin, D.: Composition of human flows and parameters of their movement during evacuation: monograph. Academy of GPS EMERCOM of Russia (2016)
29. Samoshin, D., Slyusarev, S.: Movement parameters of children with limited mobility to determine the estimated time of evacuation from buildings with their mass stay. *Fire Explos. Hazard* **25**(4), 43–51 (2016)
30. Shukla, D., Patel, E.: Speed determination of moving vehicles using Lucas-Kanade algorithm. *Int. J. Comput. Appl. Technol. Res.* **2**, 32–36 (2012). <https://doi.org/10.7753/IJCATR0201.1007>



Real-Time Information Technology Human Detection Using Cloud Services

Natalya Sokolova^(✉) , Yuliia Zhuravlova , Oleksandr Mushtat ,
and Yevhen Obydennyi 

Dnipro University of Technology, Dnipro, Ukraine
n.olegowna@gmail.com

Abstract. The current work proposes a complex solution for real-time human detection based on data received from CCTV cameras. This approach is based on modern deep learning methods and the use of cloud services such as AWS Rekognition, Cisco Meraki, MQTT broker. The solution consists of three parts: processing the video stream, human detection in the frame, mapping human locations and calculating the distance between individuals. The technology offers a semi-automatic method for remote acquisition of video stream data using modern cloud services, video stream framing, object recognition in received frames, separating human figures from other image objects, as well as counting people in the room and alerting the system administrator about exceeding the permissible limits for the presence of people. This technology has been integrated into the gym CRM-system. Testing of the modified CRM-system demonstrated its practical value and expanded capabilities for tracking and occupancy control without human intervention. Introduction of cloud technologies and machine learning in the CRM-system not only simplifies tracking and occupancy monitoring and could reduce unwanted contacts between people during the pandemic.

Keywords: Human detection · Deep learning · Object recognition · Cloud services · Human density control

1 Introduction

For the last fifteen years several pandemics have affected the world: atypical pneumonia (SARS-1, 2002), bird flu (virus of H5N1, 2003), swine influenza (virus of A/H1N1, 2009), Ebola virus (EVD, 2013). Published sources report that swine influenza caused most of deaths but the largest damage (1.5 trillion of dollars) to the world economy was inflicted by the bird flu pandemic [29]. However, the real challenge for humanity was COVID-19 (SARS-2 Cov, 2019). World Health Organization reports more than 520 million people became ill by May, 2022 [3], and World Bank reports that a world economy fell down by 6,3% in 2020. Global of growth is expected to decrease from 5.5% in 2021 to 4.1% in 2022 and 3.2% in 2023 due decreased demand and as fiscal and monetary support wind off in many

countries [15]. Quarantine restrictions diminished and/or froze economic activity of enterprises, resulting in decreased use of services, reduced sales and profits, which led enterprises reassess economic situation and future prospects. However, the losses are unevenly distributed. In particular, in Ukraine, small companies suffered more than the large ones from Covid-19 [12], and particularly ones associated with catering of mass gatherings of people - in tourism, transport, restaurant business, sports [14].

Even before the pandemic began, there was a growing interest in attracting software to business process automation and increasing importance of electronic commerce, and the recent pandemic has strengthened the trend even more. Many Ukrainian companies had to change business models to ensure order fulfillment and service delivery through digital platforms and computer systems. As a result, computer technologies became the key to providing continuity of business operations [16]. CRM-systems (Customer Relationship Management system) help automate work of enterprises through control of all communications with customers and automation operations [10].

Latest Research Papers and Article Review. Throughout a pandemic, gyms, retail chain, cinemas had to comply with quarantine restrictions, especially ones related to social distancing. Existing CRM systems, such as NetHunt CRM, Zoho, OneBox and others, did not have the function of human recognition and did not count visitors entering the buildings, which would not allow real-time estimation of occupancy rates for each room to be compliant with regulation on maximum capacity per surface area for COVID-19 pandemic. All of the above highlights the need for image recognition research.

To maintain social distance and calculate the number of people in the room was offered purely hardware solutions that use sensors to determine the number of people [7]. This solution is not always appropriate because the presence of sensors located in the room on the floor prevents the free movement of people, which in a gym or shopping center can be dangerous.

In [13] discusses a method for identifying human in an image using a pre-trained Spotfire cloud service model using Python scripts and AWS recognition models. The authors [26] propose a method of recognizing a human using information obtained from a body image made by two different types of cameras, including a visible light camera and thermal cameras. This technique reduces the effects of noise, background and changes in the appearance of the human body. A convolutional neural network (CNN) is used to extract image features. CNN is also used in [28] to identify human.

One of the most common methods of human detection in an image, including frame from CCTV cameras, is the use of computer vision techniques, including OpenCV tools with Deep learning technologies [5, 8, 9].

2 Materials and Methods

Recognition. Image recognition is the most difficult problem in the domain of computer vision that requires high processing speed and real-time recognition

accuracy. General recognition tasks include localization of object and the relationship between characteristic features. Traditional methods of pattern recognition are based on image features, descriptions of forms and image texture comparison [7, 10]. Recently, convolutional neural networks (CNN) have been most often used to recognize objects in an image [5, 26, 28]. The most effective recognition is achieved with Viola-Jones method, which can be used in real time. The latter method uses Haar features - a set of black-and-white rectangular masks of different forms - which are laid on different parts of image [8, 9, 22]. Neural networks with Deep learning technologies are also used for object recognition in real time used [21, 24, 30].

Deep Learning. Deep learning, a branch of machine learning and artificial intelligence is nowadays considered a core technology in currently ongoing Fourth Industrial Revolution [20]. Deep learning has assisted decision making in the following tasks [4, 17, 23]:

- symbols, characters recognition.
- data processing of remote sensing.
- quality control of products in manufacturing industry and agriculture.
- medical image analysis and diagnosis.
- biometrical identification of a person.
- geoanalysis for creating 3D surface maps.
- ecological monitoring of pollution levels.
- security and supervision.

Cloud Services. Object recognition requires large processing power. This may explain why Cloud services are often used in recognition in modern research [11, 19]. They allow the use of additional resources speeding up calculation. Popular Cloud services are represented with MQTT broker, Cisco Meraki, and Amazon Elastic Compute Cloud [1, 2].

MQTT broker gathers data from remote locations through a “broker” which requires a small code size though it is limited with a carrying capacity of a transmission channel. Cisco Meraki is a user-friendly and rapid development service, that provides easy user-installed network solutions that could be managed centrally through the cloud. Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, scalable computing resources in a cloud. EC2 offers rich options for virtual software development, the fastest processors and is the only cloud space with Ethernet bandwidth up to 400 Gbps. Other options include the most powerful GPU-based solution for machine learning and graphical workload courses, as well as the lowest-cost options for each logical instance output in the cloud. Copies of EC2 can be placed in different locations depending on the availability zones and regions, ensuring the continued operation of the system in case of failure in one region. Amazon EC2 high-performance computing clusters are designed to provide high bandwidth when needed for certain network-related operations.

Video Stream Processing. The video stream can be conveniently processed using the open source computer library OpenCV or AWS Rekognition [6,27]. OpenCV allows for a very fast method of human detection using histogram of oriented gradients. The method assumes that appearance and form of object can be described through a distribution of intensity gradients or directions of the edges. The image is divided into small connected cells, and a histogram of the gradient directions is compiled for the pixels in each cell. The accuracy improves when a local histogram is normalized for contrast - calculating the measure of intensity in a larger area of the image and then applying the value to normalize all cells in the block [25]. AWS Rekognition (Amazon Rekognition) uses scalable deep-learning to simplify image and video analysis with labels to identified objects and scenes [6]. The technology detects human objects on a video stream, in real time, and this could be especially helpful for the pandemic period.

3 Experiment

Information technology consists of the following steps:

- Getting frames from the video stream of CCTV cameras.
- Importing frames into the cloud-based software application.
- Objects Classification in the scenes.
- Human objects recognition.
- Calculating the distance between Human objects (requires inputs for maximum and minimum distance constants).
- Sending a message to the CRM-system about the violation of the allowable distance (when comparing the calculated distance with the threshold values of certain constants).

Functional structure of information technology implementation is graphically presented on Fig. 1 and described below:

- MV cameras are connected to the Internet via access points provide the MQTT video stream to the MQTT Meraki broker in the cloud Python script requests the URL of the image from the MV camera at intervals of few seconds.
- Meraki cloud returns a URL, and a Python script sends URL to the AWS Rekognition service.
- AWS Rekognition downloads a snapshot by URL and recognizes objects (people) based on previously trained machine learning models Recognition results are returned to the Python script and uploaded to the MQTT broker along with a snapshot URL.
- The MQTT broker forwards the results, image capture URL and other information to the Node-Red graphical user interface.

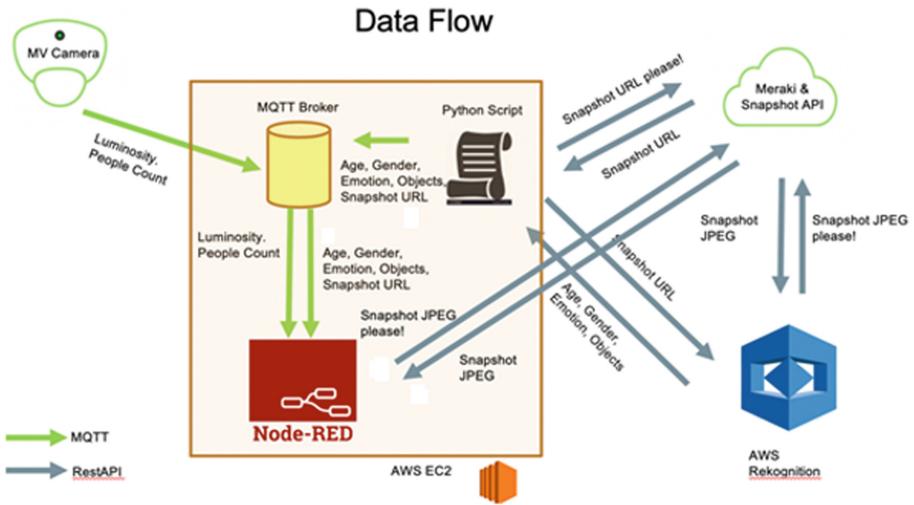


Fig. 1. Functional structure of Information Technology software implementation

The location of the cameras can affect the tracking of the client’s location, as well as the performance of wireless communication in general. Therefore, it is recommended that the cameras not be too close to each other or too far apart. The cameras subscribe to the MQTT broker and connect to the server. To access the images, you periodically request the cameras to check the serial number and corresponding URLs. After the scan, the processed frames are sent to the server for object recognition. Data is transferred to the server according to the agreed secure configuration.

The solution is implemented on the Meraki Cloud platform. To do this, Meraki access points were installed and connected to a central cloud management system using reliable Internet communication channels. A patented light tunnel with AES256 encryption is used to communicate with the Meraki Cloud. In the tunnel itself, Meraki uses HTTPS and protocol buffers for a secure and efficient solution limited to 1 kbps per device when the device is not actively managed. The Node-Red application is running in the background, working with it in terminal mode. The application is pre-configured using a configuration file in JSON format.

Learning the Recognition Model. Create and training the model to recognize objects is part of the solution and implemented with AWS Rekognition (Amazon Rekognition) - an image and video analysis service for scalable deep learning technologies to create models for recognizing people’s faces, text, events, unacceptable content, and customizable labels, object and scene identification (including human figures). The development of the model includes Amazon Rekognition Custom Labels for recognized and identified specific objects and consists of the following steps:

- Step 1. Create an image collection.
- Step 2. Define your classes.
- Step 3. Create datasets for training and testing.
- Step 4. Marking of classes on each image of datasets.
- Step 5. Define classes in the dataset.
- Step 6. Model training.
- Step 7. Validation of the trained model.
- Step 8. Improving the trained model if necessary.
- Step 9. Deploy the model in the cloud service.

Training requires a project with a dataset of training and test data. Training data are used in the model learning cycle. Test data are used to assess the performance of the model and determine how applicable the trained model is to unstudied data.

For the training procedure, a training dataset of 70 images from CCTV cameras (including those that are freely available on the Internet) and a test set of 30 images were created.

To determine the location of objects in the image, a model is created using bounding boxes (a rectangle that isolates the object in the image). The object is identified by assigning a label to the bounding box. A border should be created on each image that contains objects. Training and test images contain a bounding box that surrounds various objects in the image and a label that identifies humans.

At the next stage, image classes are defined with help of labels (each class corresponds to a label at the image level), and the list of classes (human, equipment) is compiled. Image level labels apply for images in both training and testing data sets.

A JSON string created for each image contains a link source field that indicates the location of the image, and contains selected bounding boxes for each object in the image. The JSON string set is a manifest file. After locating an object in the model, a JSON string is updated in the manifest files. Next, image-level labels are assigned to training and test datasets using a hierarchical taxonomy of ancestral labels to classify labels - one image level label for each image in the training and testing datasets. An image level label is a class that represents each image.

To learn the model, Amazon Rekognition Custom Labels creates a copy of the original training and test images. By default, copied images are encrypted with a key. Model learning is based on Deep learning technologies.

In the process of learning, the learning outcomes for individual test images were adjusted. After the preliminary evaluation, the trained model was improved. Gradient optimization (gradient of the loss function to minimize the number of errors) was used to iteratively refine the model parameters.

Evaluation of productivity and accuracy of the trained model was carried out on a test set. For each image in the test set, we compared predictions of the learned model with the assigned label, using common metrics for evaluating

the effectiveness of the machine learning model as Recall, Precision and F1-score (confidence/best score) [18]. Amazon Rekognition Custom Labels returns metrics for test results for the entire test data set, as well as metrics for each special label.

During testing, Amazon Rekognition Custom Labels application predicts whether a test image contains a special label. Confidence indicator is a value that quantifies the confidence of the model. If the confidence level for a custom label exceeds a threshold, the model will include that label. For each image we used true positive, false positive and false negative indicators. These prediction results were used to calculate scores for each label, as well as aggregates for the entire test suite. Same definitions apply to predictions made by the model at the constraint level, with the difference that all metrics are calculated for each constraint (prediction or baseline truth) in each test image.

Amazon Rekognition Custom Labels automatically calculates the estimated threshold (0–1) for each of the custom labels. The predicted threshold for each label is the value above which the prediction is considered true or false. The set of values is based on a test dataset. The estimated threshold is calculated based on the best result F1-score (1) achieved in the test data set during model training.

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2 \frac{precision \times recall}{precision + recall} \quad (1)$$

Amazon Rekognition Custom Labels uses F1-score to measure the average performance of a model for each label and the average performance of a model for the entire test set. Model performance is an aggregate measure that takes into account both Recall and Precision for all labels. Results of the preliminary evaluation of the model are shown in Table 1. After preliminary evaluation, additional learning was conducted on 20 more images to improve productivity. For the trained model, Amazon Rekognition Custom Labels can find a trade-off between Recall and Precision using the MinConfidence input parameter for DetectCustomLabels. The higher the MinConfidence value, the higher the Precision is but Recall is lower. At a lower value of MinConfidence we get a high Recall, but lower Precision. Results of the evaluation of the improved model are shown in Table 2.

Table 1. Initial training results

Metric	Value
Rrecision	0.927
Recall	0.750
F1-score	0.829

After training, the model is ready for deployment to a specific address and subsequent recognition. For each user, we created an access key to create credential files and to configure credentials in the AWS command line interface on the

Table 2. Initial training results

Metric	Value
Rrecision	0.985
Recall	0.847
F1-score	0.910

host. Calculation of room occupancy. As a result of executing the script after object recognition, we obtained a data structure that contains information about humans detected by the camera in a room. The number of records in the data structure are assumed to represent the number of human who were found in the frame. If the number of human exceeds the maximum occupancy of the room (set by a predefined constant in the configuration file of the script), a message is sent to the CRM system.

If the number of records is more than 1 and does not exceed the allowable norms, the distance between the detected human in the frame is calculated. The calculation is based on the specified initial data “MIN_DISTANCE”, “MAX_DISTANCE” and “THRESHOLD” (minimum allowable, maximum allowable distance and limit of the total number of violations, respectively), which are set in the configuration file for the script. To calculate the distance, each selected figure is linked to the reference points on a grid defined for the room. After calculating the distances, results are visualized by a rectangular contour of detected humans with a circle in the center. For convenience, the data are recorded on top of the frame in a text view, and duplicated in the program console. In the original image (Fig. 2), the system classifies objects and complements the text, which describes the behavior of objects in the frame (Fig. 3).

**Fig. 2.** Image obtained from the camera stream, without application processing

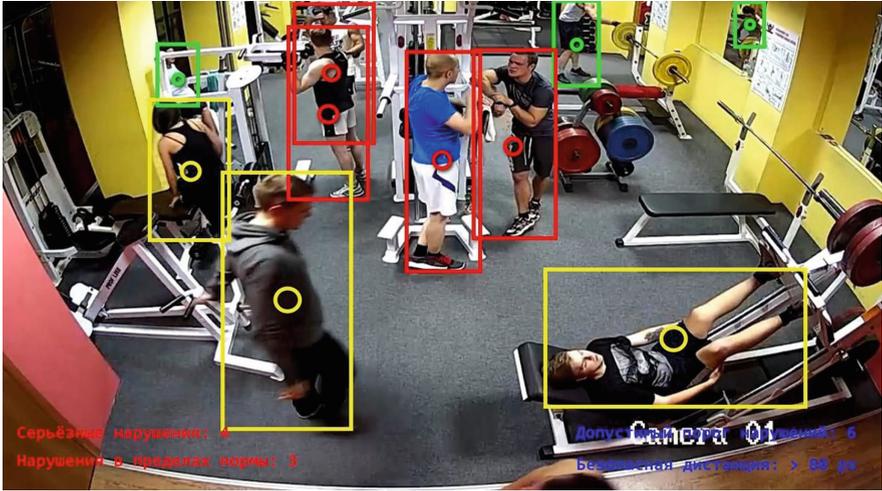


Fig. 3. Image describing the behavior of objects in the frame

The application compiles the following information on the image:

- information field on the number of serious violations;
- information field on the number of violations that fall within normal limits;
- information field on the number of permissible serious violations at the time of capture;
- information field of safe distance between objects.

When the number of violations exceeds the allowable threshold (e.g., exceeding the limit on number of violations in the frame, Fig. 4), the application sends messages to the CRM console and emails to the administrator (Fig. 6) using the library “Mailer”, which allows send a message to a provided mail address.

The developed application was tested in the fitness center SportLife in Dnipro, Ukraine. The following equipment was installed: 4 Cisco MV72 Camera; 4 Cisco Meraki MR34 access points; 1 Cisco MS320 access switch; Cisco MS420 distribution level switches; Cisco Meraki MX400 Firewall (Fig. 6). The cameras were placed at the corners of the halls for a wider field of view. This allowed not only to cover most of the space, but also to save on the total number of cameras. Overall, we installed 4 cameras in different areas of the fitness club. Camera locations allowed the coverage of areas with the highest rate of movement.

4 Results and Discussion

The current work solves a complex problem of tracking people in a room in real time using frames from CCTV cameras. Developed a model successfully recognized human figures in the image. The model used Deep learning technologies

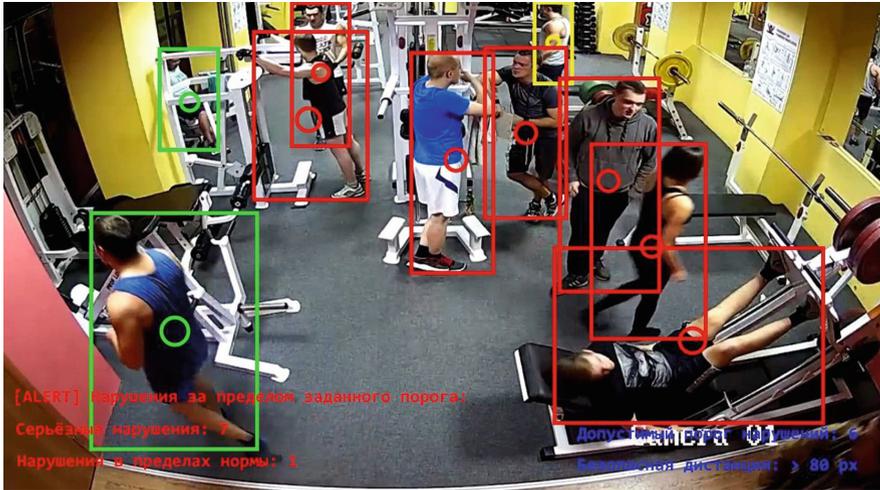


Fig. 4. Image describing the behavior of objects in the frame and the text of exceeding the permissible limit of violations

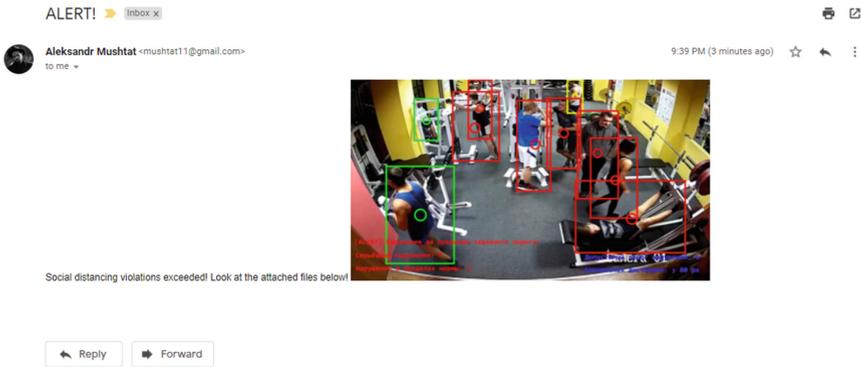


Fig. 5. Example of an e-mail notification about exceeding the specified limit of permissible violations

on a special dataset, evaluated using common metrics and improved using a test dataset.

The introduction of cloud technologies and machine learning in the automated system not only simplifies the tracking and control of the filling of the territory, but also expands business opportunities.

The practical value of the results is that this system was built with a few available technologies expands the capabilities of the standard process of tracking objects and automated delivery.

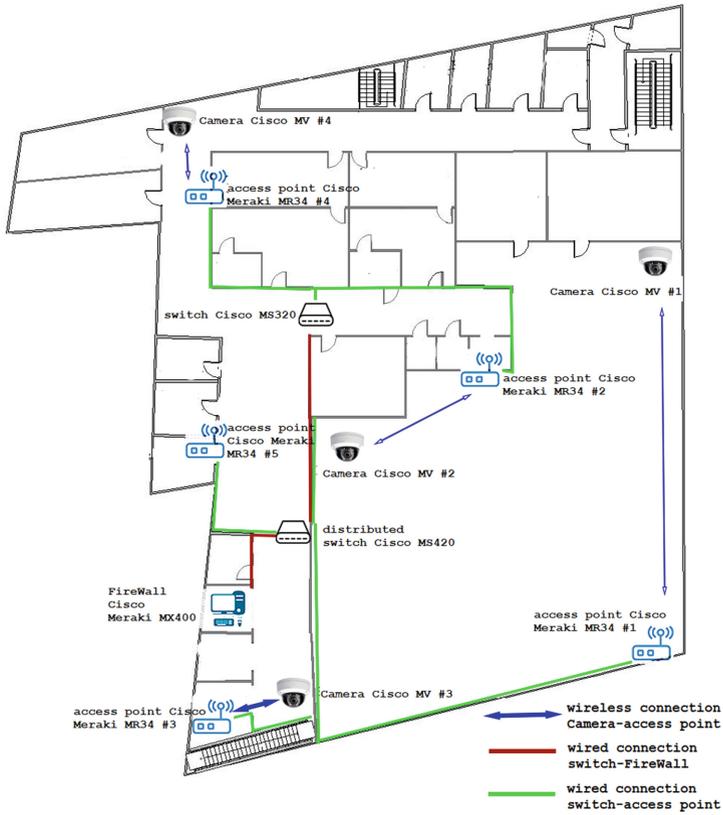


Fig. 6. Scheme of equipment placement on the territory of the fitness center “Sportlife”

5 Conclusions

Deep learning remains the most promising approach in Computer Vision. The combination of modern hardware, Cloud services and software tools allow to develop and implement applications that automatically recognize real-time human figures in video streams, calculate the distance between people and inform in case of exceeding maximum occupancy rate in the room or violating social distancing. The method allows to create safe conditions for visiting gyms, shopping malls, cultural institutions, and the administration of such institutions to monitor compliance with quarantine restrictions, minimizing contact between staff and visitors.

Further areas of research are to obtain additional information about the visitor (gender, age, mood). This information about visitors may allow business or institution to improve infrastructure, logic and the business process overall through conducting analytics of visit and assuring the quality of services. As a service to customers, an algorithm could be developed in popular communication

tools, social applications or messengers in order to inform customers about the number of people on the premises of the institution in real time.

References

1. Cisco meraki. official page. <https://meraki.cisco.com/>
2. Cloud computing services - amazon web services (aws). https://aws.amazon.com/ru/?nc1=h_ls
3. Covid-19 coronavirus pandemic. <https://www.worldometers.info/coronavirus/>
4. Deep learning in ArcGIS pro—ArcGIS pro j documentation. <https://pro.arcgis.com/en/pro-app/help/analysis/image-analyst/deep-learning-in-arcgis-pro.htm>
5. How to detect people using computer vision. <https://learn.alwaysai.co/object-detection>
6. Machine learning image and video analysis - amazon rekognition - amazon web services. https://aws.amazon.com/rekognition/?nc1=h_ls
7. People density control solution. <https://www.wg-plc.com/product/people-density-control-solution>
8. Real-time human detection in computer vision. <https://medium.com/@madhawavidanapathirana/https-medium-com-madhawavidanapathirana-/real-time-human-detection-in-computer-vision-part-1-2acb851f4e55>
9. Real-time human detection with OpenCV. <https://thedatafrog.com/en/articles/human-detection-video/>
10. Shcho take CRM-sistema: povnyi hid po vyboru CRM. <https://nethunt.ua/blog/shcho-takie-crm-sistema-povnyi-ghid-po-vyboru-crm-dlia-pochatkivtiv-v-2021/>
11. Steve ranger. an introduction to cloud computing, right from the basics up to IaaS and PaaS, hybrid, public, and private cloud, Aws and azure (2022). <https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/>
12. Vplyv pandemii covid 19 ta karantynnu na profesiinu diialnist. <https://eba.com.ua/vplyv-pandemiyi-covid-19-ta-karantynu-na-profesijnu-diyalnist/>
13. Colin gray. image recognition in tibco spotfire® using Python and Aws (2019). <https://community.tibco.com/feed-items/image-recognition-tibco-spotfirer-using-python-and-aws>
14. Hromadska orhanizatsiia “tsentr prykladnykh doslidzhen” predstavnytstvo fondu konrada adenauera v ukraini “vplyv covid-19 ta karantynnykh obmezhen na ekonomiku ukrainy” (2020). <https://www.kas.de/documents/270026/8703904/Vplyv+COVID-19+ta+karantynnykh+obmezhen+na+ekonomiku+Ukrainy.+Kabinetne+doslidzhennia+TsPD.+Lypen+2020.pdf/b7398098-a602-524d-7f88-6189058f69d3?version=1.0&t=1597301028775>
15. Global growth to slow through 2023, adding to risk of ‘hard landing’ in developing economies, 11 Jan 2022. <https://www.worldbank.org/en/news/press-release/2022/01/11/global-recovery-economics-debt-commodity-inequality>
16. Androshchuk, H.: Efekt pandemii covid-19: tsyvrovi tekhnolohii — kliuch do rozvytku biznesu. <https://yur-gazeta.com/golovna/efekt-pandemiyi-covid19-cifrovi-tehnolohiyi--klyuch-do-rozvitku-biznesu.html>
17. Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W.J., Ducoffe, M.: Potential, challenges and future directions for deep learning in prognostics and health management applications. *Eng. Appl. Artif. Intell.* **92**, 103678 (2020). <https://doi.org/10.1016/j.engappai.2020.103678>

18. Furnkranz, J., (ed.): An analysis of ruleevaluation metrics. In: Conference: Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), 21-24 Aug 2003, Washington, DC, USA (2003)
19. Hnatushenko, V.V., Sierikova, K.Y., Sierikov, I.Y.: Development of a cloud-based web geospatial information system for agricultural monitoring using sentinel-2 data. In: 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). vol. 1, pp. 270–273 (2018). <https://doi.org/10.1109/STC-CSIT.2018.8526717>
20. Hnatushenko, V., Zhernovyi, V.: Method of improving instance segmentation for very high resolution remote sensing imagery using deep learning. In: Babichev, S., Peleshko, D., Vynokurova, O. (eds.) DSMP 2020. CCIS, vol. 1158, pp. 323–333. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61656-4_21
21. Höfener, H., Homeyer, A., Weiss, N., Molin, J., Lundström, C.F., Hahn, H.K.: Deep learning nuclei detection: a simple approach can deliver state-of-the-art results. *Comput. Med. Imaging Graph.* **70**, 43–52 (2018). <https://doi.org/10.1016/j.compmedimag.2018.08.010>
22. Hordiiuk, D., Oliinyk, I., Hnatushenko, V., Maksymov, K.: Semantic segmentation for ships detection from satellite imagery. In: 2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO), pp. 454–457 (2019). <https://doi.org/10.1109/ELNANO.2019.8783822>
23. Hordiiuk, D., Hnatushenko, V.: Neural network and local laplace filter methods applied to very high resolution remote sensing imagery in urban damage detection. In: 2017 IEEE International Young Scientists Forum on Applied Physics and Engineering (YSF), pp. 363–366 (2017). <https://doi.org/10.1109/YSF.2017.8126648>
24. Huang, J., Shang, Y., Chen, H.: Improved Viola-Jones face detection algorithm based on HoloLens. *EURASIP J. Image Video Process.* **2019**(1), 1–11 (2019). <https://doi.org/10.1186/s13640-019-0435-6>
25. Navneet, D., Bill, T.: Histograms of oriented gradients for human detection. In: INRIA Rhoˆne-Alps, 655 avenue de l’Europe, Montbonnot 38334, France (2005)
26. Nguyen, D.T., Hong, H.G., Kim, K.W., Park, K.R.: Person recognition system based on a combination of body images from visible light and thermal cameras. *Sensors* **17**(3), 605 (2017). <https://doi.org/10.3390/s17030605>
27. Nixon, M., Aguado, A.: Feature Extraction and Image Processing for Computer Vision. 4th Ed., vol. 627, Academic Press (2020)
28. Shuiwang, J., Xu, W., Yang, M., Kai, Y.: 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(1), 221–231 (2013). <https://doi.org/10.1109/TPAMI.2012.59>
29. Vasilyeva, T., Lyeonov, S.: Covid-19, SARS, H5N1, A/H1N1, EVD: porivnialnyi analiz vplyvu pandemii na ekonomichniy ta sotsialnyi rozvytok u natsionalnomu, svitovomu ta rehionalnomu kontekstakh. *Naukovyi pohliad: ekonomika ta upravlinnia*, **3**(69) (2020). <https://doi.org/10.32836/2521-666X/2020-69-4>
30. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I–I (2001). <https://doi.org/10.1109/CVPR.2001.990517>



Deep Learning Technology for Automatic Burned Area Extraction Using Satellite High Spatial Resolution Images

Vita Kashtan^(✉)  and Volodymyr Hnatushenko 

Dnipro University of Technology, Dnipro, Ukraine
vitalionkaa@gmail.com, vvgnat@ukr.net

Abstract. Today, there is an acute issue of prompt provision of up-to-date and most complete spatial information for making optimal management decisions in the forestry industry. The most effective tool for solving many problems in this area is using satellite data. The advent of optical satellites of high spatial resolution (Sentinel-2, Worldview-2,3) makes it possible to use modern methods of operational mapping to solve forestry problems (in particular, identifying burnt areas after a fire). As the monitoring area increases, and the number of images increases, the need for automatic data processing. Thereof to improve the accuracy of recognition of such objects in images is to use deep learning algorithms based on convolutional neural networks. This paper presents a description of the “traditional” methods for burned area detection (vegetation indices, multi-channel, and single-channel change detection, etc.), convolutional neural networks, and their main principles, and limitations. We also present a new method for operational mapping of a burned area using convolutional neural networks and provide an increase in the accuracy of forest fire recognition by more than two times in comparison with traditional methods based on raster arithmetic. The accuracy, estimated by the F-measure, is 0.80 for Haciveliler, 0.93 - Mugla, and 0.95 - Kavaklidere.

Keywords: Machine learning · Burned are · Segmentation · High spatial resolution · Convolutional neural networks · Vegetation indices

1 Introduction

Timely, and prompt detection of violations in the forest fund is an important task in the structure of forestry. Every year, a significant area of forests is exposed to adverse factors: fires, windfalls, snow breaks, ice, outbreaks of pests, forest diseases, as well as anthropogenic factors. All these factors lead to the complete death or partial drying of trees. However, the most serious threat to forests is fire.

Forest fires are one of the most common disasters of both natural, and human nature. This is an almost unmanageable process, more than one hectare of natural plantations is destroyed. This leads to the loss of animals and plants. In

addition, forests are the main sink of carbon gas and greenhouse gas emissions on the planet Earth. Therefore, to avoid such situations, it is necessary to turn to modern methods for analyzing forested areas.

Satellite imagery is actively used to track and detect negative changes in forests. The task of detecting objects of interest, and anomalies in satellite images has been relevant for more than a decade, and the number of applications where it arises is only increasing. Among them are applications related to remote sensing of the Earth [7,15], ground monitoring [20,27], radar [10], agriculture [22,30], and many others.

Moreover, in contrast to traditional methods of collecting information, satellite images have the following advantages: coverage of a larger area, analysis of the quantitative characteristics of the earth using radiometrically calibrated sensors, and synoptic reviews of events about the environment.

To disturb the forest cover, several prior information is used up to a given library of standards, which makes it possible to identify them on satellite images. These include spectral image, shape, area, internal structure, and texture. In addition, this allows you to detect a certain area of the image, the characteristics of which differ from the predicted ones formed during image processing - an anomalous area (malignant tumors, the appearance of artifacts, etc.). One of the main signs of disturbances in forest vegetation is the spectral image. A feature of vegetation is a relatively small reflection in the red region of the spectrum, and a greater reflection in the near-infrared (NIR). With damage to plants and a decrease in photosynthesis, the reflection in the red region of the spectrum increases, while in the near-infrared zone it decreases. In addition, the reflectivity in the average NIR increases significantly, which is associated with a drop in the moisture content of damaged trees. The more significant changes in the spectral reflectance, the easier it is to identify damaged stands [33].

2 Problem Statement

In recent years, the number of new satellites has been increasing. They provide real-time monitoring with high spatial resolution, and detail, and conduct schematics in the middle, near-IR channel with a lower spatial resolution (multispectral images). Existing deep learning methods (detection of changes in images) developed for Landsat data that do not apply to Sentinel-2 data.

The aim of current work is to develop new automated technology for operational mapping of forest cover disturbances (burned area), taking into account spectral (visible, and near-IR range of the spectrum), geometric (the significance of which increases with an increasing spatial resolution of the data) features based on convolutional neural networks. This will make it possible to automatically recognize the area of burnt forest cover in high spatial resolution images.

The method must consider the unordinary nature of high-resolution satellite imagery, structure, features violations of the forest cover (shape, area, etc.), and data for training convolutional neural networks. The article's contribution lies

in the complexity of the proposed approach, and the coverage of all aspects of detecting forest cover disturbances, including the geometric model of objects interest, and setting up the neural network architecture for training the fire detection from satellite images.

3 Literature Review

The image dimension (monitoring area) is the number of coordinates that describe each pixel (for example, spatial coordinates, spectral ranges), the larger this dimension, the more difficult it is to implement these tasks in practice. Therefore, there is a need to automate this process. Methods for automating the detection of violations based on these signs have been developed for several decades, almost since the launch of the first satellites of the Landsat series and the appearance of the first digital images of the Earth from space.

Gounaridis et al. [6] used multitemporal Landsat satellite images from 2008 to 2011 and GLSDEM data to map Greek land cover types at the national spatial coverage level in 2010. We used the random forest (RF) algorithm and a set of 7 zonal images, 10 index images calculated from zonal images, and 3 geographic features for the image analysis. The overall accuracy was 83%.

Mohamed et al. [26] proposed using multitemporal Landsat satellite images acquired over the period 2010–2018 to detect land cover type change in Syria at the national spatial coverage level using a post-classification method. The maximum likelihood algorithm (ML) and a set of 3 zonal images were used to interpret satellite images. Fragou, et al. [8] used multitemporal Landsat satellite images acquired over the period 1993–2010 to detect land cover type change in Greece at the local level of spatial coverage using the post-classification method. To interpret satellite images, the SVM algorithm and a set of 6 zonal images and 8 zonal image ratios were used. Subsequently, the NDII index was shown to be highly effective for detecting areas of forest damage by pests [9, 11, 28]. Sriwongsitanon et al. [35] proposed using NDII as a proxy for soil moisture in the root zone. Testing of this methodology in 8 sub-basins of the Upper Ping River Basin in Thailand showed the effectiveness of the NDII index. It is consistent with the study by Almeida et al. [3] who found reasonable correlations between Landsat 7 NDII values and measured moisture content in the root zone of rained olive trees in Tunisia, supporting the use of NDII as a proxy for soil moisture content in arid regions.

The same work confirmed that NDID is more effective in detecting forest canopy disturbances than the well-known NDVI. In addition, the NBR difference index is very effective for identifying burnt areas and has a high correlation with the degree of forest damage by fires [23]. Achour et al. [13] applied dNBR-based indices for Sentinel-2, and Landsat-8 satellites using Tunisia as an example. This led to the conclusion that the images of the Sentinel-2 satellite proved to be slightly more effective than the Landsat-8 data for mapping. To detect forest cover disturbances, some work has considered the use of more complex indices than simple normalized differences. These include both indices based on the

visible and near-IR ranges of the spectrum (SAVI, and others), and using the mid-IR range, as well as the results of the Tasseled Cap transformation [1]. The Tasseled Cap transformation is a special case of Principal Component Analysis. In [12], a comparative analysis of various vegetation indices to effectiveness, and several other methods for detecting changes in Landsat images was carried out. Using the damage of Hurricane Katrina in the USA as an example, it was found that this transformation (Tasseled Cup) provides the greatest accuracy. This conversion uses 6 channels of the TM/Landsat imaging system with different weights, with the maximum weight having the middle infrared channel. Huo L.-Z [17] proposed to conduct the separation of all types of forest cover disturbance in the United States using a modern, and efficient random forest classifier. However, the accuracy of the method turned out to be low. To identify infringements more precisely, it is necessary to take into account not only spectral characteristics.

The main limitation of all the above methods based on spectral features is the impossibility of automated recognition of the types of forest cover disturbances (clearings, burnt areas, windblown, damage by pests, and diseases). To solve this problem, additional analysis of geometric, and textural features is required, without which the separation accuracy will be low. Therefore, a new qualitative solution for the automatic detection of forest disturbances has been achieved with the advent of deep learning, and convolutional neural networks CNN [16].

The neural network adjusts the neuron weights when processing the available observations, and choosing the decision threshold. The use of convolutional networks allows for finding objects of different sizes, and classes. In this case, at least two options for their application are possible. In some tasks, readymade or pre-trained CNN architectures can be successful using. CNN can recognize only those objects on which they were trained. For many more specific tasks, you can use a new database of images, and objects to further train a model with a pre-trained architecture (transfer learning) for a specific narrow task [4, 5]. In the second case, the weights are usually adjusted only for the last fully connected layers; all other coefficients of the model do not change, since they serve in the network to extract features from the image.

The development of neural network detectors for detection tasks has led to the emergence of much architecture, in particular, deep neural networks [25]. The main sector of their development was effective real-time detected. The Region-Based (R-CNN) network, which is aimed at recognizing objects, not in the entire image, but in local areas of images, can be especially singled out [36]. The R-CNN architecture has spawned a series of domain-specific improved models, including the Fast R-CNN network [38] for the task of classifying, and regressing a rectangle covering an object; network Faster R-CNN [32], using an auxiliary subnet to generate regions of interest.

Networks of the You Only Look Once architecture [31] demonstrate even greater performance, in one pass forming, and limiting the regions of the localization of the object, and the label of the object class. High speed, and performance are also provided by the Single Shot Multibox Detector network [24, 39], which is based on the discretization of the output space of rectangular detection

areas into rectangles from a standard set with specified sizes for each location on the feature map (characteristic features) of the image.

In 2020, the results of works on the wind blow recognition in the forests of Western Europe [34], and clear-cuts in Ukraine [21] using convolutional neural networks of the U-net architecture were published. The accuracy of identifying the consequences of wind blows was 40–50%, for cuttings, the accuracy turned out to be slightly higher - about 50–55%. Of the other algorithms based on machine learning, there is an experience in using the Random Forest classifier to recognize forest power disturbances from Landsat images [14]. The detection accuracy was about 90% while missing objects was about 8%, and the proportion of false positives was 20%, which can be considered a successful result.

In general, the review, and analysis of the literature shows that deep learning methods are developing quite quickly, which leads to the obsolescence of some algorithms. The main limitation of existing methods of forest cover disturbance types based on deep learning is a large training set (dataset). In this case, these methods are used to mask cloudiness or classify land-use types. And at the same time, the existing methods are less studied in practice for violations of the forest cover. Therefore, the problem of detecting forest cover disturbances using these algorithms has a high level of novelty.

4 Materials and Methods

In this paper, we introduce a deep learning technology for automatic burned area extraction using satellite high spatial resolution images (see Fig. 1). Using CNN help to learn segmentation tasks jointly from multispectral, and panchromatic images. The proposed technology is presented in Fig. 1 and consists of the following main steps:

1. Loading input satellite images for post-fire dates.
2. Preprocessing and geometric correction of satellite images.
3. Pansharpening.
4. Spectral index calculation and burn mapping.
5. Deeping learning model for burned detection.

In the following sub-sections, we present the methodology used within the framework of our research.

4.1 Satellite Data

The first step in the process of identifying forest cover disturbances is to select and upload images that cover the required area. The proposed technology in this work analyzes only a couple of images to detect forest cover disturbances and does not take into account the time aspect.

In a rapid detection and mapping of forest cover disturbances, the most suitable data are from satellites of high spatial and temporal resolution.

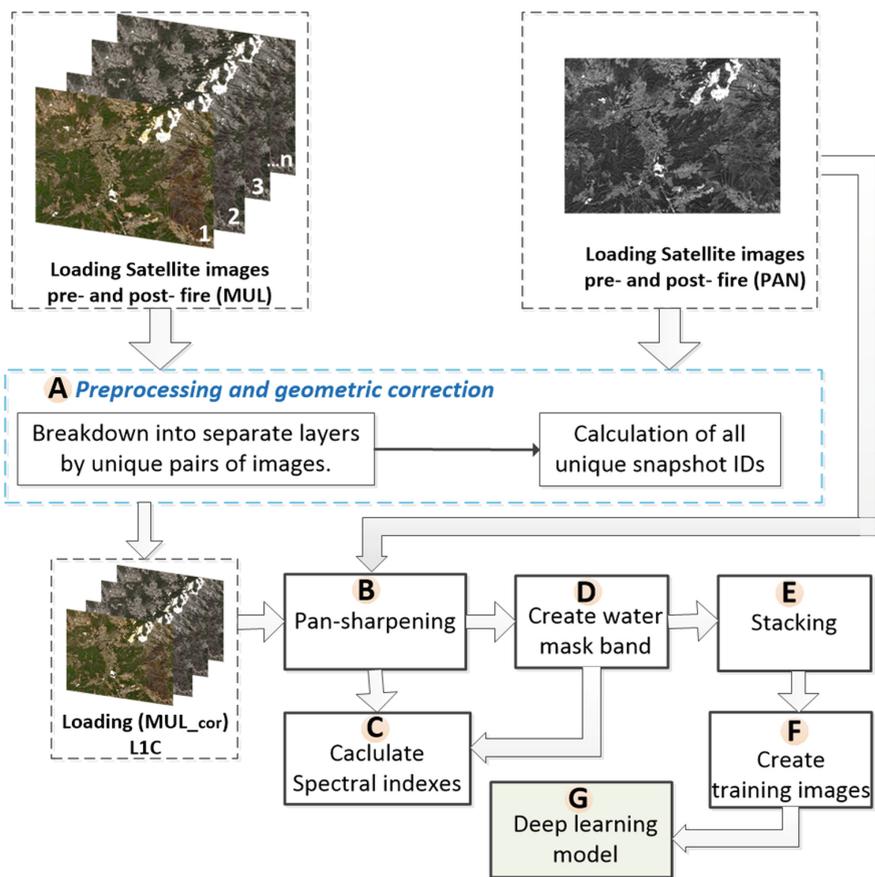


Fig. 1. Flowchart of suggested burned area algorithm

The Sentinel-2 (A/B) spacecraft are equipped with a multispectral Imager opto-electronic multispectral sensor for imaging with a resolution of 10 to 60 m in 13 spectral channels in the visible, near IR, and shortwave IR zones of the spectrum.

WorldView-2 is equipped with a telescope with a 110 cm aperture, which makes it possible to obtain digital images of the earth’s surface with a spatial resolution of 0.46 m in the panchromatic range and 1.84 m in 8 multispectral ranges when shooting at nadir. The eight-channel high-resolution spectrometer provides imaging in the traditional spectral zones: red, green, blue, and near infrared-1 (NIR-1), as well as in four additional spectral zones: violet (coastal - coastal), yellow, red edge, and near infrared-2 (NIR-2). This ensures that differences in the state of vegetation are displayed, including changes in time, and also minimizes the effect of the atmosphere on image quality. The uniqueness of the Sentinel-2 mission is associated with a combination of large territorial coverage, frequent resurveys, and, as a result, the systematic acquisition of full

coverage of the entire Earth by high-resolution multispectral imaging. Images from this device are distributed free of charge [18]. In this paper, we used satellite images of high spatial resolution before and after the fire. The forest fires in the Kavaklidere, Mugla, and Haciveliler (Fig. 2) regions in 2021 (from July 2021 to October 2021) were chosen as thematic fires for this example.

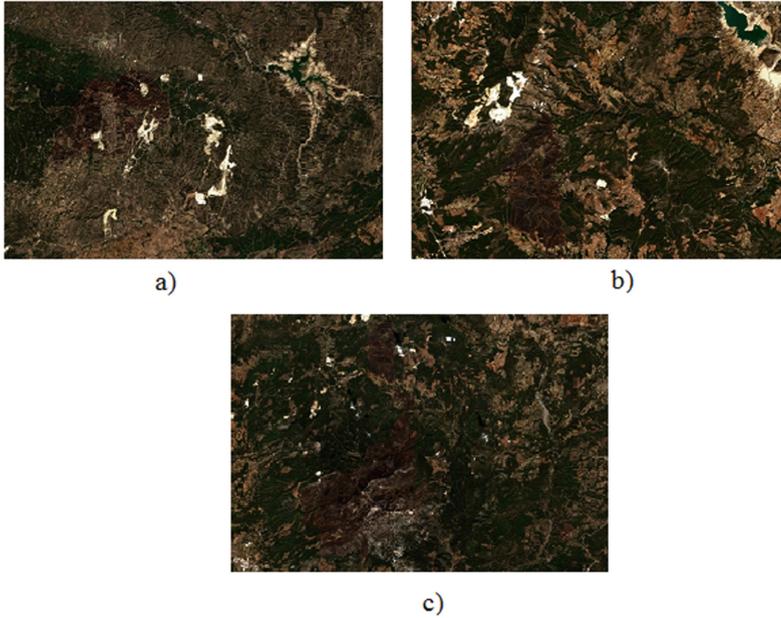


Fig. 2. Satellite images by Sentinel-2 of Turkey: a) Haciveliler; b) Kavaklidere; c) Mugla

4.2 Preprocessing

Satellite images of a fixed object (scene), obtained in different spectral intervals, have different spatial and radiometric differences and differ significantly in spatial distributions of brightness. However, each such image has a separate informational significance in terms of the characteristics of the object (scene). Therefore, in the second stage, an atmospheric correction was carried out based on a set of tools for preliminary processing of satellite images and the creation of a unified coverage of the area of interest. Radiometric calibration corrects pixel values. This is to correct for instrument distortion which may cause an error in quantitative physical measurements of satellite images. The equation removes both illumination and atmospheric effects [19]:

$$R = DN \times slope \pm intercept, \quad (1)$$

where DN is the digital numbers of bands.

Next, we perform geometric correction of data for pre-processing of satellite images and eliminate geometric distortions for the individual elements of the image (pixels) in the appropriate planimetric coordinates. Mathematically, the geometric correction can be represented as a polynomial. This will correspond to the global coordinates of the distorted image and the coordinates of the reference image. Polynomial models are usually needed to convert the coordinates of the image into an object [19]:

$$\begin{aligned} X &= a + a_1x + a_2y + a_3xy, \\ Y &= b + b_1x + b_2y + b_3xy, \end{aligned} \tag{2}$$

where x, y are the new coordinates; a, b are the satellite images.

The result is classified as rasters. The conversion was done using the gd library. The obtained classification results were translated into vector form (shp format). Loading images (level L1C).

4.3 Pansharpening and Spectral Index Calculation

After the image has undergone the pre-processing described above, the low-resolution multispectral image is merged with a higher-resolution panchromatic raster layer. This process is also known as “pansharpening”. In the last two decades, a large number of satellite image fusion methods have been developed such as HSV, ICA, Color Normalized Brovey, Grama-Schmidt, PC Spectral Sharpening, and combined methods. But according to the work [19], the separate use of existing methods for increasing the spatial resolution of multichannel images contains the appearance of spectral distortions. Therefore, it is proposed to use the method based on HSV-converting wavelet transform and hyperspherical color conversion, which allows not only improves the spatial resolution of the primary digital satellite image but also avoids the spectral distortion [8,26,28]. Figure 3 shows the results of the proposed in [19] pansharpening methods.

The NBR, dNBR, and BAIS2 were performed as spectral indices and calculated for a forest fire. It’s determining the location of the fire. The Normalized Burn Ratio has been used to watch changes in vegetation caused by a fire in the near-infrared range [2]:

$$NBR = \frac{NIR - SWIR2}{NIR + SWIR2}, \tag{3}$$

where: NIR is the near-infrared band; $SWIR2$ is the longer shortwave infrared.

In the next step, the NBR difference (dNBR) between the pre-fire image and the post-fire image gave an idea of the intensity of the combustion:

Another coefficient Burned Area Map Spectrum Index has been specifically designed to take advantage of the spectral properties of a satellite image.

$$dNBR = NBR_{prefire} - NBR_{postfire}. \tag{4}$$

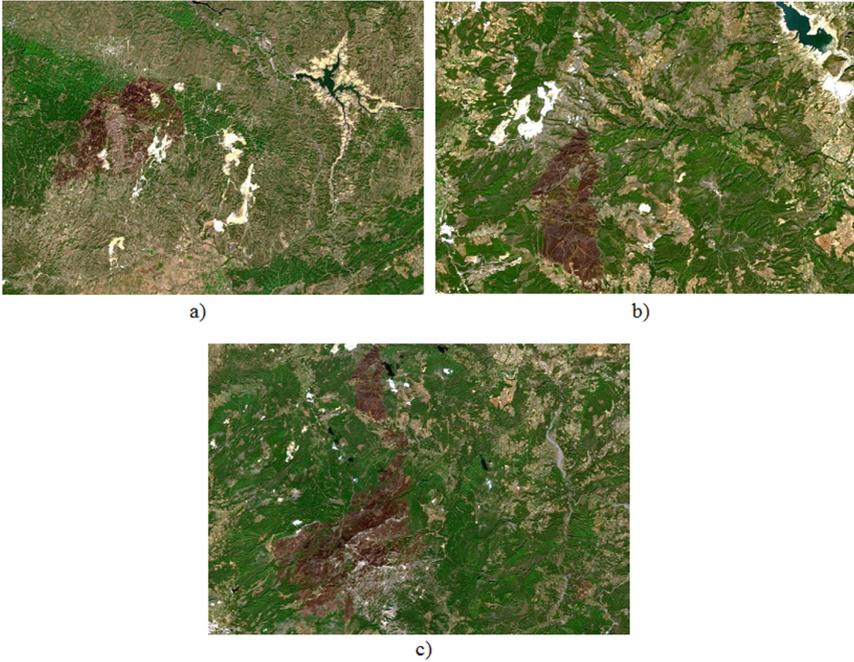


Fig. 3. Results of pansharpening method: a) Haciveliler; b) Kavaklidere; c) Mugla

This index uses the characteristics of vegetation described in the red edge spectral domains and the radiative response in the shortwave infrared [2]:

$$\begin{aligned}
 BAIS = 1 - & \left(\frac{\sqrt{Red_{ed1} + Red_{ed2} + NIR_n}}{Red} \right) \\
 & - \left(\frac{SWIR2 - NIR_n}{\sqrt{SWIR2 + NIR_n}} + 1 \right), \tag{5}
 \end{aligned}$$

where Red_{ed1} is the shorter red edge, Red_{ed2} is the longer red edge, Red is the visible red band, $SWIR2$ is the longer shortwave infrared band, NIR_n is the vegetation red edge band.

The proposed technology checks for the presence of imagery satellite bands and then proceeds to calculate the indices. This helps to choose the best combination for the spectral indices of the forest fire. Figure 4 shows the result of applying spectral indexes of the satellite image of the Kavaklidere region. The information of the spectral channels allows for determining the violations in the vegetation. A feature of vegetation is a relatively small reflection in the red channel of the spectrum and more in the near-infrared channel. The burnt area is displayed in white. Besides, it's shown us that the area of the river is highlighted in white which is unacceptable for recognizing burnt areas.

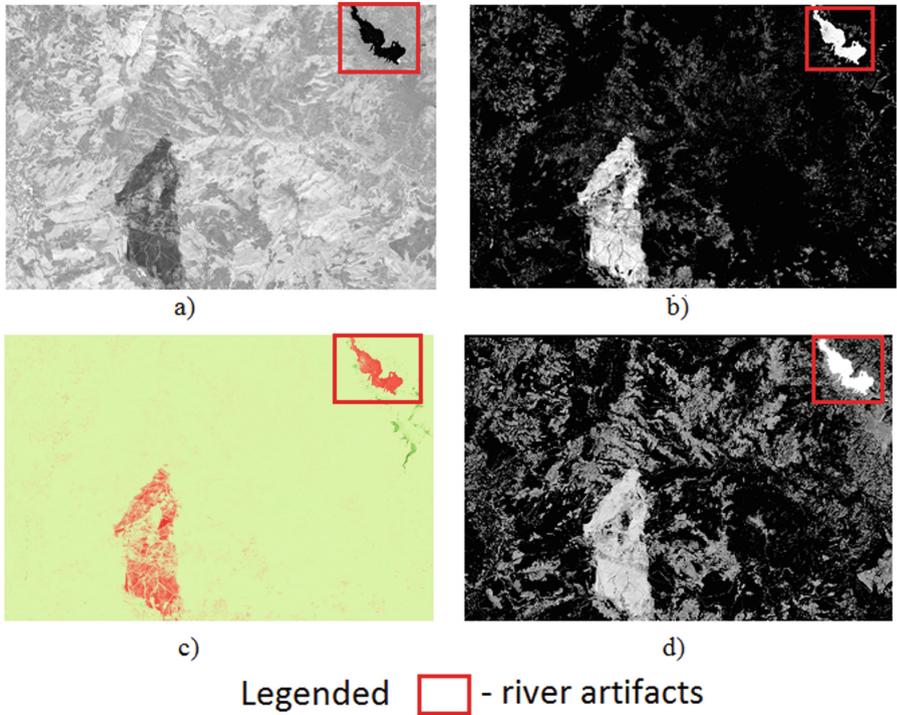


Fig. 4. Satellite image of the area of Kavaklidere showing: a) NBR; b) dNBR; c) NBR in color; d) BAIS

The next step (D step according to Fig. 1) is to create a water mask to remove the dark areas due to water spectrum absorption based on the Modified Normalized Water Index (MNDWI). The MNDWI spectral index uses a short-wave infrared band instead of near-infra. It's improving the accuracy of water detection due next ended with a non-parametric detection of a local threshold [9]:

$$MNDWI = \frac{\rho_{green} - \rho_{swir1}}{\rho_{green} + \rho_{swir1}} \tag{6}$$

So, water Pixels were masking from the images by applying the following equation:

$$W = \left(\frac{(NIR_n + SWIR1 + SWIR2) - Coas_e + Blue + Green}{(NIR_n + SWIR1 + SWIR2) + (Coas_e + Blue + Green)} \right) < 0, \tag{7}$$

where NIR_n is the vegetation red edge band, $SWIR1$ is the shorter short-wave infrared band, $SWIR2$ is the longer shortwave infrared band, $Coas_e$ is the coastal aerosol band, $Blue$ is the visible blue band, $Green$ is the visible green band.

Figure 5 shows the result of applying spectral indexes of the burns search into account the index of the water. These results shows that, the satellite image after applying the spectral indices eliminates the artifacts associated with water pixels.

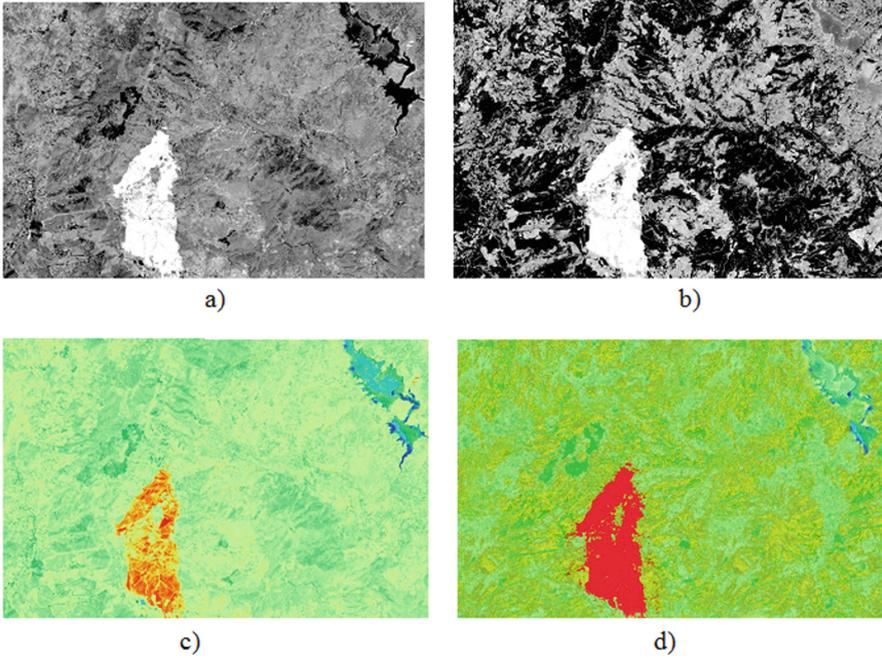


Fig. 5. Satellite image of the area of Kavaklidere showing: a) NBR; b) dNBR; c) NBR in color; d) BAIS

After all, these steps had processing to create a single coverage of the area of interest (stacking). So, for multispectral images (red-, green-, blue-, red edge-, NIR-, NIR narrow-, SWIR) bands were resampled to panchromatic resolution. Then a spectral index calculation was applied with the water mask definition. After Layer stacking, bands of the selected satellite images were standardized using the following equation [29]:

$$(Bi \leq 0) * + (Bi \geq 10000) * 1 + (Bi \geq \text{and} Bi < 1000) * \text{float}(Bi) \setminus 10000, \quad (8)$$

where Bi is the Specific Band.

4.4 Deep Learning Model

Building machine learning models effective requires an appropriate set of training data. In this work, to train the neural network, it was necessary to present the

resulting training sample in the form of tiles - small raster images, which contain the target mask and a set of input features. In training models to be able to calculate various derived features (for example, vegetation indices) the original brightness values of all channels of both images were stored in the tiles.

For each pair of images in the training set, the channels of the first and second images were reduced to a spatial resolution of 10 m and combined into one 26-band raster. Further, the violation layer corresponding to this pair was converted into a raster format and added as an additional channel to the created raster. Thus, for each pair of images, a 27-channel raster was obtained, which contains a mask of violations and brightness values in 13 channels of images obtained before and after the appearance of a violation).

The next step is using convolutional neural networks (CNN) in the task of detecting fires on satellite images. CNN consists of convolution layers, subsampling (pooling), and fully connected layers (fully connected). The convolutional layer is the purpose to combine the brightness of pixels in the local neighborhood for the subsequent extraction of common features characteristic of the image. The convolution procedure is performed by a sliding square window of small size ($3 \times 3, 5 \times 5, 7 \times 7 \dots$ pixels), the so-called convolution kernel. It implements a weighted summation of pixels within a local area (which “combines” the brightness), and the resulting sum characterizes the feature corresponding to the local area of the image that fell into the convolution kernel. As a result of the movement of such a window over the entire image (for example, from left to right and from top to bottom), a matrix (feature map) is obtained, where weighted sums of pixels from local neighborhoods will be presented at individual points. Figure 6 shows the model of automatic burned extraction based on the UNET architecture.

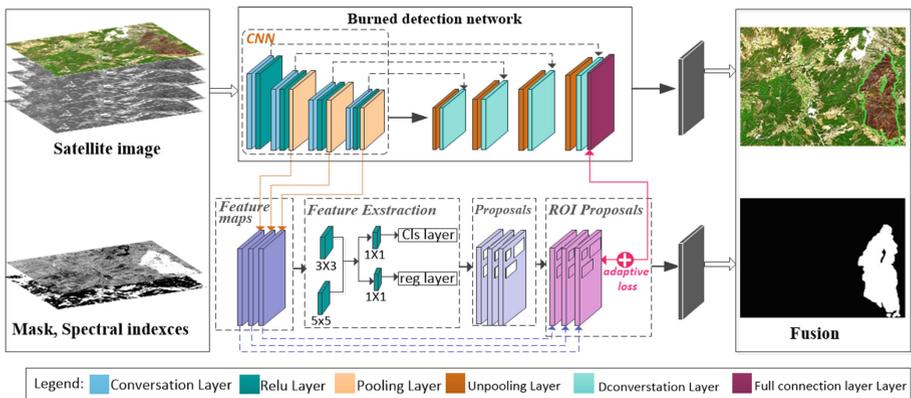


Fig. 6. Flowchart technology of automatic burned detection

At the same time, two networks are implemented in UNET: fire detection and a network that directly performs image segmentation based on the Faster

Region-based Convolutional Network (R-CNN). R-CNN searches for objects not on the entire image, but on preselected areas (regions) where objects of interest may be present. So, for fires (our object interest), sharply torn borders are characteristic, the shape is always elongated by the prevailing wind direction [38]. On the other hand, the information in the spectral channels makes it possible to determine disturbances in the vegetation. A feature of vegetation is a relatively small reflection in the red channel of the spectrum and more in the near-infrared channel (the greater the change in spectral reflectance, the easier it is to identify damaged plantings). Therefore, these data must be taken into account when training our network. Image feature extraction is based on the VGG16 network model. This approach consists of using the weights of an already existing model (transfer learning) and allows highlighting the basic geometric features in the satellite images. Such a network is used as a “preprocessor” of images, highlighting low-order geometric features on it. Next, we add a rectified linear unit (ReLU) activation function and perform layer fusion [38] to extract the features of the input satellite image. To convert features from different regions to given sizes, a layer of combining regions of interest (Region of Interest Pooling) was proposed. To do this, the region area is divided into a grid with cells of size:

$$H/h \times W/w, \tag{9}$$

where H and W are the sizes of the region, h and w are the sizes of the union core.

Every group is followed by a max-pooling layer for its sampling, with an $h \times h$ kernel (in this work we propose 2×2). Adding another module Region Proposal Network (RPN) to the R-CNN network allows for to calculation of regions based on the received feature map (see Fig. 7).

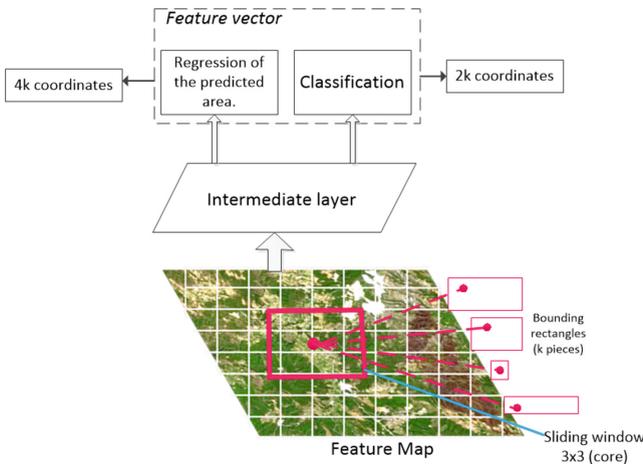


Fig. 7. Region suggestion method (RPN) based on anchor boxes

The RPN module processes (Fig. 7) the extracted features of the entire image based on a simple neural network. Outputs from such a neural network are transmitted in parallel for processing by two fully connected layers. The first layer is designed to assess the linear regression of the geometric parameters of the region (the similarity of objects with burnt areas), and the second one is responsible for the direct classification of objects in the region. The outputs of such layers use anchor boxes to make bounding rectangles predictions with different sizes for each position of the sliding window. The first layer for each anchor box determines 4 coordinates to refine the position of the object area. The second layer produces two numbers that determine the probability of the presence or absence of an object in the region. This allows excluding “uninteresting” regions. Both layers are trained simultaneously. The loss function is given as the sum of the loss functions for each layer with weight coefficients. The regions left behind are passed to a processing unit implemented following the Fast R-CNN architecture. Thus, we get a special network that offers regions for research.

The learning step plays an important role in the realization of our technology. For the proposed technology in this work such a complex architecture not to lose the ability to detect low-level features (individual lines, angles, etc.), training was carried out at a low learning step of 0.00001 and 0.001 (Fig. 8). It also decreased with every 20th iteration. The lower value of the training step allowed the network to train longer with the preservation of the original features.

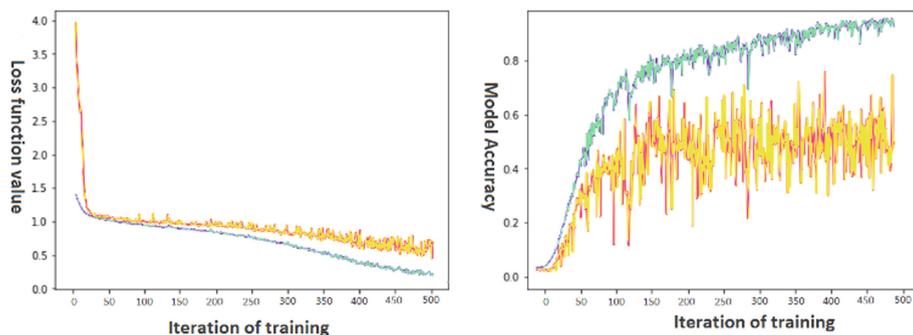


Fig. 8. Network training schedule with preprocessor with the initial value of the learning step 0.001

5 Experiment

Three areas of applied research were chosen for our study, they are Kavaklidere, Mugla, and Haciveliler located in Turkey. The following factors influenced the choice of our study area, firstly, they experience severe forest fires from time to time, which have a massive impact on people’s lives and the deterioration

of the economy and the environment. Therefore, it is important to find out the causes of the fire and take measures to mitigate future fires, as well as for the permanence of the forest ecology. Secondly, they have a forest structure that quickly affects the spread of fire. Thirdly, there are recent forest fires that may be of interest to our study.

5.1 Visual Analysis

Satellite images (from Sentinel-2 presented in Fig. 9) before and after the fire were selected for visual analysis, which allowed visual interpretation of the data. High spatial resolution satellite data is used to map charred and unburned areas, and access the characteristics of unburned areas and their spectral properties. This is since they contain information about backscattering, which is sensitive to the structural parameters of the forest. Figure 1 shows the primary stage of a forest fire, which is difficult to identify fires in this image. The analysis of the proposed technology shows according to Fig. 9, accurate and reliable information about the position and length of unburned areas, as well as vegetation or other soil cover contained in unburned areas.

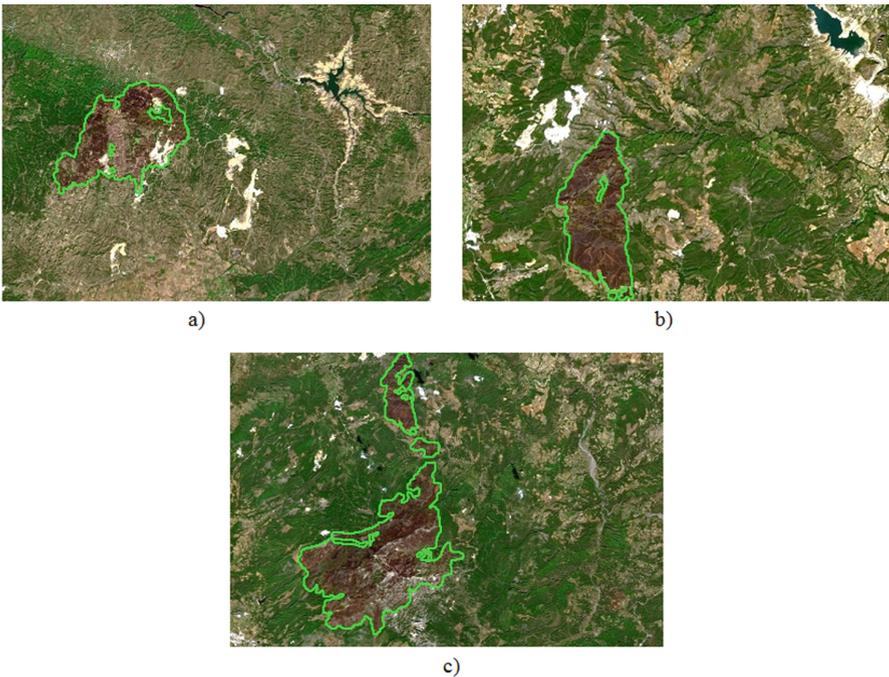


Fig. 9. Burned detection of satellite image high resolution: a) Haciveliler; b) Kavaklidere; c) Mugla

5.2 Quantitative Analysis

Evaluation of efficiency in solving problems of detecting forest cover disturbances should include information about the presence of an object in the image (classification), as well as determining its location (localization). Also, in a dataset where the class distribution is uneven, a simple accuracy-based metric will introduce variances. It is also important to assess the risk of misclassification. Thus, it is necessary to link the confidence score and correlate the predicted bounding boxes with the true ones. The most common metrics for evaluating burned area detection results are precision (PB) and recall (RB). PD is the proportion of correct predictions relative to their total number [37]:

$$PB = \frac{TP(c)}{TP(c) + FP(c)}, \quad (10)$$

where PB is the precision of burned area; $TP(c)$ is True Positive predication of the presence of an object of class C , and in fact, there was an object of class C ; $FP(c)$ is False Positive predication the presence of an object of class C , and in fact, an object of class C did not exist.

RB measures how well the detector finds all positive samples. For example, find 80% of all possible positive predictions [37]:

$$RB = \frac{TP(c)}{TP(c) + FN(c)}, \quad (11)$$

where RB is the recall of burned area; $TP(c)$ is the True Positive predication of the presence of an object of class C and in fact, there was an object of class C ; $FN(c)$ is False Negative predication the absence of an object of class C , and in fact, an object of class C existed.

A total of 45 control objects (burnt areas) were analyzed. In the first stage, for each control object, the presence of an intersection between its area and the areas of changes identified using the algorithm was determined (in this case, the threshold for the probability of changes was taken equal to 10%). If the area of intersection between the control and detected object was more than 10% of the area of the control object, then such a control object was considered successfully detected, and the rest of the objects were evaluated as not detected. After that, all objects from the layer predicted by the model that intersects with the control ones were removed. The remaining objects (which did not intersect with the control ones) were visually analyzed to determine the presence/absence of disturbance of the forest cover in this location.

Thus, four indicators were obtained: the total number of control objects, the number of selected objects from the control, missed objects, and false positives. All the experiments were run with Vidia GeForceGTX 1050 GPU accelerators and 8-GB GPU memory.

6 Results and Discussion

To implement the proposed technology for the automated determination of forest fires has been developed to a graphical user interface which is present in Fig. 10. It includes a Menu bar and Toolbar. The Toolbar includes Open, Burn Detection, Pre-processing, and Exit for quick access to the functions in menu bar items. The Menu bar includes File, Actions, Help, and Exit.

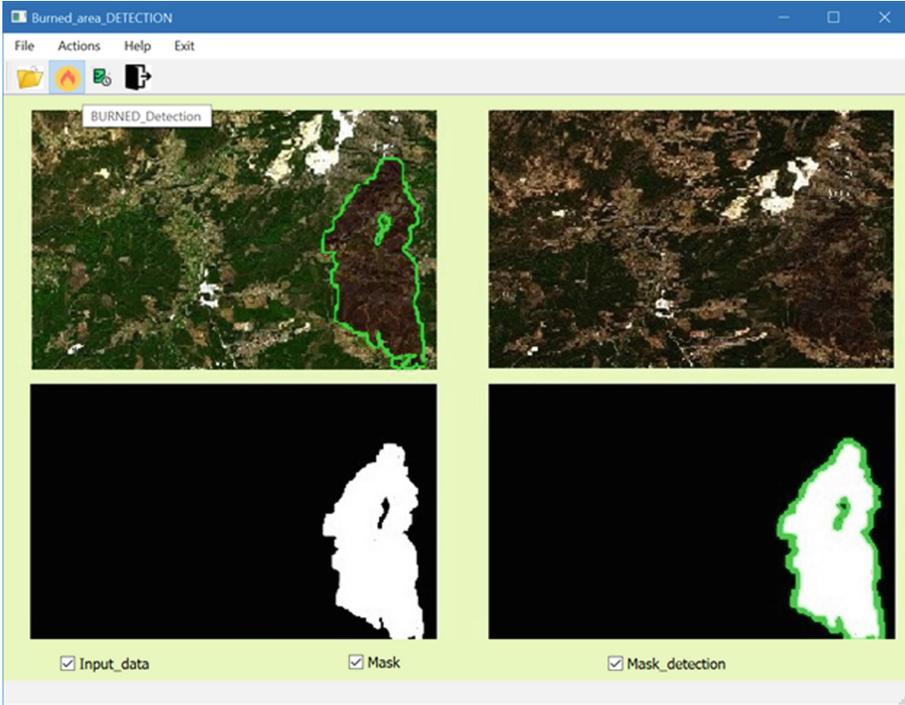


Fig. 10. User friendly graphical interface of proposed technology

Table 1 and Table 2 presents results on the image from described satellite images in comparison to traditional CNN methods and proposed deep learning technology for burned detection.

Table 1. Percentage of detected objects in control zones

Method	Haciveliler			Kavaklidere			Mugla		
	TP	FP	FN	TP	FP	FN	TP	FP	FN
Proposed, %	92.6	4.1	4.8	91.6	2.6	3.9	94.8	5.4	7.7

Intersection over Union (IoU) measures the overlap between two regions. Figure 11 shows how the IoU works. This value is used to determine the percentage of overlap between the true and predicted areas of the object [39]:

$$IoU \approx \frac{TP}{TP + FN + FP}. \tag{12}$$

Table 2. Percentage of detected objects in control zones

Method	Haciveliler			Kavaklidere			Mugla		
	PB	RB	IoU	PB	RB	IoU	PB	RB	IoU
Proposed	0.96	0.95	0.91	0.97	0.95	0.93	0.95	0.92	0.88
U-Net	0.94	0.90	0.85	0.90	0.86	0.77	0.91	0.83	0.76
CasNet	0.95	0.91	0.87	0.88	0.82	0.72	0.89	0.80	0.71

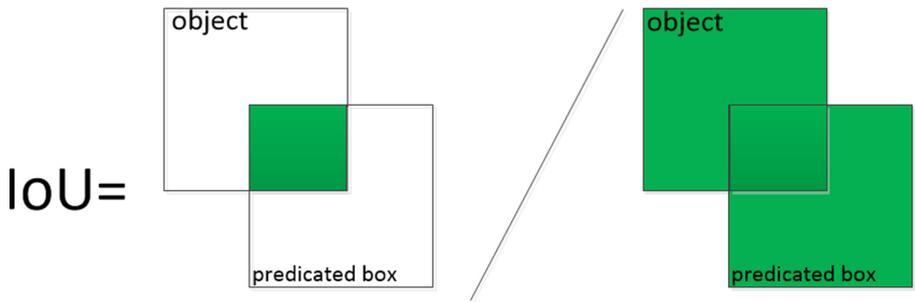


Fig. 11. Calculating of IoU

In all cases, the proportion of burned objects identified by the model was over 92% (max 96%). The share of missed objects averaged 4.2%, and false positives - 5.4%. It should be noted that false alarms occur most often in riverbeds and floodplains, due to fluctuations in the water level and the associated sharp changes in brightness characteristics. Therefore, the exclusion of these areas from the analysis, taking into account the water mask, reduces the frequency of false errors. The Mugla region is characterized by the minimum accuracy of clearcut selection from the control set (88%). This is due to the complex geometry of forest fires. Besides proposed determining the degree of spectral contrast between the burnt and unburnt areas. This was done by assessing the use of the f-measurement coefficient [2]:

$$F - measure = \frac{2 \times PB \times RB}{PB + RB}. \tag{13}$$

Table 3. Classification results for f-measurement coefficient after proposed technology for satellite images

Haciveliler F-measure	Kavaklidere F-measure	Mugla F-measure
0.80	0.95	0.93

Table 3 consists of the results of the f-measurement coefficient after proposed technology.

It showed the capability of both the spectrum index and polarimetry in corresponding to changes in vegetation structure after a fire. All fire zones yielded an accuracy average of 0.89.

7 Conclusions

In the current work, a new automated technology for operational mapping of forest cover disturbances (burned areas) using convolutional neural networks is suggested. The review of the scientific literature and existing analogs confirmed the relevance of the task. The theoretical part of convolutional neural networks (principles of their operation, information about convolution layers, and sub-samples) is also considered. An image preprocessing process was formulated.

During the analysis, the following subtasks were formulated: perform processing of satellite images, conduct pansharpening, highlight the area after the fire in the image, create a water mask and analyze the information obtained using a convolutional neural network. All these subtasks are implemented in the form of a Python program. It has been established that the deep learning-based algorithm (CNN) provides more than a twofold increase in recognition accuracy compared to traditional methods. CNN detects selective burns that stand out as integral objects while using traditional methods it is possible to select only individual pixels with the largest changes in brightness. The accuracy of recognition of forest cover disturbances by the proposed technology averaged 92%; F-measurement, is 0.80 for Haciveliler, 0.93 - Mugla, and 0.95 - Kavaklidere.

The developed models can be used in monitoring systems with different time resolutions (weekly, monthly, quarterly, annual).

Acknowledgments. We thank the European Union Copernicus program for providing Sentinel data accessible via the Copernicus Open Access Hub.

References

1. Abate, N., Elfadaly, A., Masini, N., Lasaponara, R.: Multitemporal 2016–2018 Sentinel-2 data enhancement for landscape archaeology: the case study of the Foggia Province, Southern Italy. *Remote Sensing* **12**(8), 1309 (2020). <https://doi.org/10.3390/rs12081309>

2. Al-hasn, R., Almuhammad, R.: Burned area determination using Sentinel-2 satellite images and the impact of fire on the availability of soil nutrients in Syria. *J. Forest Sci.* **68**, 96–106 (2022). <https://doi.org/10.17221/122/2021-JFS>
3. Almeida, O., et al.: Validation of normalized difference infrared index (NDII) to estimate soil moisture in traditional olive cultivation systems, Tunisia. In: EGU General Assembly Conference Abstracts, p. 14209. EGU General Assembly Conference Abstracts (2019)
4. Alzubaidi, L., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **8**(53), 1–74 (2021). <https://doi.org/10.1186/s40537-021-00444-8>
5. Asifullah, K., Anabia, S., Umme, Z., Aqsa, S.: A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **53**, 1–62 (2020)
6. Dimitrios, G., Anastasios, A., Sotirios, K.: Land cover of Greece, 2010: a semi-automated classification using Random Forests. *J. Maps* **12**, 1055–1062 (2016). <https://doi.org/10.1080/17445647.2015.1123656>
7. Ding, C., et al.: Boosting few-shot hyperspectral image classification using pseudo-label learning. *Remote Sens.* **13**(17), 3539 (2021). <https://doi.org/10.3390/rs13173539>
8. Fragou, S., Kalogeropoulos, K., Stathopoulos, N., Louka, P., Srivastava, P., Karpouzias, E.: Quantifying land cover changes in a mediterranean environment using landsat TM and support vector machines. *Forests* **11**(7), 750 (2020). <https://doi.org/10.3390/f11070750>
9. Gao, H., Hrachowitz, M., Sriwongsitanon, N., Fenicia, F., Gharari, S., Savenije, H.: Accounting for the influence of vegetation and landscape improves model transferability in a tropical savannah region. Delft University of Technology (2016). <https://doi.org/10.1002/2016wr019574>
10. Garkusha, I., Hnatushenko, V., Vasyliiev, V.: Research of influence of atmosphere and humidity on the data of radar imaging by Sentinel-1. In: 2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO), pp. 405–408 (2017). <https://doi.org/10.1109/ELNANO.2017.7939787>
11. Giorgos, M., Ioannsi, M., Irene, C.: Evaluating and comparing Sentinel 2a and Landsat-8 operational land imager (OLI) spectral indices for estimating fire severity in a Mediterranean pine ecosystem of Greece. *GIsci. Remote Sens.* **55**, 1–18 (2018). <https://doi.org/10.1080/15481603.2017.1354803>
12. Goswami, A., et al.: Change detection in remote sensing image data comparing algebraic and Machine Learning methods. *Electronics* **11**(3), 431 (2022). <https://doi.org/10.3390/electronics11030431>
13. Hammadi, A., Ahmed, T., Hichem, T., Wahbi, J.: Evaluation and comparison of Sentinel-2 MSI, Landsat 8 OLI, and EFFis data for forest fires mapping. illustrations from the summer 2017 fires in Tunisia. In: Geocarto International, pp. 1–20 (2021). <https://doi.org/10.1080/10106049.2021.1980118>
14. Hethcoat, M., Edwards, D., Carreiras, J., Bryant, R., França, F., Quegan, S.: A machine learning approach to map tropical selective logging. *Remote Sens. Environ.* **221**, 569–582 (2019). <https://doi.org/10.1016/j.rse.2018.11.044>
15. Hnatushenko, V., Hnatushenko, V., Mozgovyi, D., Vasyliiev, V., Kavats, O.: Satellite monitoring of consequences of illegal extraction of Amber in Ukraine. *Sci. Bull. Natl. Min. Univ. State High. Educ. Inst. “Natl. Min. Univ.”* **2**(158), 99–105 (2017)
16. Hordiiuk, D., Hnatushenko, V.: Neural network and local laplace filter methods applied to very high resolution remote sensing imagery in urban damage detection. In: 2017 IEEE International Young Scientists Forum on Applied Physics and Engineering (YSF), pp. 1984–1988 (2017). <https://doi.org/10.1109/ysf.2017.8126648>

17. Huo, L., Boschetti, L., Sparks, A.: Object-based classification of forest disturbance types in the Conterminous United States. *Remote Sens.* **11**(5), 477 (2019). <https://doi.org/10.3390/rs11050477>
18. Jérôme, L., et al.: Sentinel-2 sen2cor: L2A processor for users. In: *ESA Living Planet Symposium 2016*, vol. SP-740, pp. 1–8 (2016)
19. Kashtan, V., Hnatushenko, V.: A Wavelet and HSV Pansharpening Technology of High Resolution Satellite Images. In: *1st International Workshop on Intelligent Information Technologies and Systems of Information Security (IntellITSi-2020)*, pp. 67–76 (2020). <http://ceur-ws.org/Vol-2623/paper7.pdf>
20. Kashtan, V., Hnatushenko, V., Shedlovska, Y.: Processing technology of multispectral remote sensing images. 2017 IEEE International Young Scientists Forum on Applied Physics and Engineering (YSF), pp. 355–358 (2017). <https://doi.org/10.1109/ysf.2017.8126647>
21. Kostiantyn, I., Mykhailo, Y., Seliverstov, V.: Deep learning for regular change detection in Ukrainian forest ecosystem with sentinel-2. *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.* **14**, 364–376 (2021)
22. Liming, Z., Jiming, L., Chen, L.: Vehicle detection based on remote sensing image of YOLOV3. In: *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (2020). <https://doi.org/10.1109/ITNEC48623.2020.9084975>
23. Liu, J., Maeda, E., Wang, D., Hesikanen, J.: Sensitivity of spectral indices on burned area detection using landsat time series in Savannas of Southern Burkina Faso. *Remote Sens.* **13**(13), 2492 (2021). <https://doi.org/10.3390/rs13132492>
24. Liu, Y., Zhong, Y., Qin, Q.: Scene classification based on multiscale Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **56**, 7109–7121 (2018). <https://doi.org/10.1109/TGRS.2018.2848473>
25. Miranda, M.S., de Santiago, V.A., Körting, T.S., Leonardi, R., de Freitas, M.L.: Deep convolutional neural network for classifying satellite images with heterogeneous spatial resolutions. In: Gervasi, O., et al. (eds.) *ICCSA 2021*. LNCS, vol. 12955, pp. 519–530. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87007-2_37
26. Mohamed, A., Anders, J., Schneider, C.: Monitoring of changes in land use/land cover in Syria from 2010 to 2018 using multitemporal landsat imagery and GSI. *Land* **9**(7), 226 (2020). <https://doi.org/10.3390/land9070226>
27. Prati, A., Shan, C., Wang, K.I.K.: Sensors, vision and networks: from video surveillance to activity recognition and health monitoring. *J. Ambient Intell. Smart Environ.* **11**(1), 5–22 (2019). <https://doi.org/10.3233/Ais-180510>
28. Quintano, C., Fernández-Manso, A., Fernández-Manso, O.: Combination of landsat and Sentinel-2 MSI data for initial assessing of burn severity. *Int. J. Appl. Earth Obs. Geoinf.* **64**, 221–225 (2018). <https://doi.org/10.1016/j.jag.2017.09.014>
29. Rahman, S., Chang, H., Magill, C., Tomkins, K., Hehir, W.: Spatio-temporal assessment of fire severity and vegetation recovery utilising Sentinel-2 imagery in New South Wales, Australia. In: *International Geoscience Remote Sensing Symposium*, pp. 9960–9963 (2019)
30. Rauf, H., Saleem, B., Lali, M., Khan, M., Sharif, M., Bukhari, D.: A citrus fruits and leaves dataset for detection and classification of citrus diseases through Machine Learning. *Data Brief* **26**, 104–340 (2019). <https://doi.org/10.1016/j.dib.2019.104340>
31. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 779–788 (2016)

32. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. MIT Press, pp. 91-99 (2015)
33. Rogers, T.W., Jzccard, N., Griffin, L.D.: Automated X-ray image analysis for cargo security: critical review and future promise. *J. X-ray Sci. Technol.* **25**, 33–56 (2017)
34. Scharvogel, D., Brandmeier, M., Wesi, M.: A deep learning approach for calamity assessment using Sentinel-2 Data. *Forests* **11**(12), 1239 (2020). <https://doi.org/10.3390/f11121239>
35. Sriwongsitanon, N., Gao, H., Savenije, H., Maekan, E., Saengsawang, S., Thianpopirug, S.: Comparing the normalized difference infrared index (NDII) with root zone storage in a lumped conceptual model. *Hydrol. Earth Syst. Sci.* **20**, 3361–3377 (2016). <https://doi.org/10.5194/hess-20-3361-2016>
36. Tahir, A., et al.: Automatic target detection from satellite imagery using machine learning. *Sensors* **22**(3), 1147 (2022). <https://doi.org/10.3390/s22031147>
37. Walter, S., Michele, M., Valter, C., Giorgia, F., Frattaroli, A.: The potentiality of Sentinel-2 to assess the effect of fire events on mediterranean mountain vegetation. *Plant Sociol.* **57**, 11–22 (2020). <https://doi.org/10.3897/pls2020571/02>
38. Yulang, C., Jingmin, G., Kebei, Z.: R-CNN-based Satellite components detection in optical images. *Int. J. Aeros. Eng.*, 10 (2020). <https://doi.org/10.1155/2020/8816187>
39. Zhu, Q., Zhong, Y., Li, D., Zhang, L.: Adaptive deep sparse semantic modeling framework for high spatial resolution image scene classification. *IEEE Trans. Geosci. Remote Sens* **56**, 6180–6195 (2018)



Classification Methods of Heterogeneous Data in Intellectual Systems of Medical and Social Monitoring

Olena Arsirii^(✉) , Svitlana Antoshchuk , Olga Manikaeva ,
Oksana Babilunha , and Anatolii Nikolenko 

Odesa Polytechnic National University, Odesa, Ukraine
e.arsiriy@gmail.com, nikolenko@ukr.net

Abstract. The research is devoted to the solution of the scientific and practical problem of creation of intelligent systems (IS) of medical and sociological monitoring (MSM) on the basis of the improved models of representation of heterogeneous MSM's data and the developed methods of their classification by means of Kohonen network training. In the research improve the models of representation of weakly structured and weakly labeled heterogeneous MSM's data in the spaces of properties and features, taking into account values, types, formats, sources, quality assessments and procedures of aggregation/transformation of properties of detailed data. Based on these models, a method of matching classes and clusters markers in learning of Kohonen network with partial teacher involvement was developed, which is based on constructing a two-dimensional histogram of pairwise matches of classes and clusters markers values with its subsequent intersecting by rows and by columns to the developed rule. The method allows to obtain additional class markers in the unlabeled part of the training sample. Based on the method of matching classes and clusters markers to assess the suitability of chromosomes in the population of the genetic algorithm for each example from the training sample, a method of heuristic weight adjustment in the learning process of the Kohonen neural network is proposed. The use of such an adjustment of the weights allowed to reduce the training time of the Kohonen network without losing the level of reliability of the classification. The method of classification of weakly structured and weakly labeled heterogeneous MSM's data has been improved due to the use of developed methods of matching class and cluster markers and heuristic adjustment in the process of learning the Kohonen network

Keywords: Intelligent systems · Heterogeneous data · Weakly structured data · Weakly labeled data · Kohonen neural network · Genetic algorithm

1 Introduction

In complicating the global epidemiological situation, special attention is paid to monitoring the living and working conditions of the target audience with the

The original version of this chapter was revised: Reference 6 has been revised. The correction to this chapter is available at https://doi.org/10.1007/978-3-031-16203-9_40

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023, corrected publication 2023
S. Babichev and V. Lytvynenko (Eds.): ISDMCI 2022, LNDECT 149, pp. 686–704, 2023.

https://doi.org/10.1007/978-3-031-16203-9_38

help of intelligent systems (IS) of medical and sociological monitoring (MSM) of the new generation [5, 9, 22]. Such IS MSM allows to collect and keeping heterogeneous data on the state of public health or microclimate of the social environment of the studied target audience, as well as to make classification decisions about lifestyle or working conditions of each of the respondents [2, 7]. Analysis of existing MSM IS, such as “MedExpert”, “Eximed”, “Meditex”, “Myrian”, “Centrak”, etc., shows that for classification solutions in such systems are increasingly using methods of teaching deep neural networks with a teacher [8]. But the practical application of such methods requires long-term pre-processing of weakly structured heterogeneous MSM’s data, which are not only collected from different sources, but also interpreted using different, not always related, qualitative or quantitative scales and often contradict each other. This fact significantly reduces the efficiency required in the conditions of monitoring or the reliability of classification decisions. In addition, in a situation where weakly structured heterogeneous MSM’s data are periodically updated and supplemented, it is impossible to obtain a sufficient number of classification solutions in the form of pre-marked data. Therefore, heterogeneous MSM’s data can also be classified as weakly labeled.

It should be noted that the classification of such heterogeneous data uses the training of neural networks with partial involvement of the teacher, which can be implemented through self-learning of the Kohonen neural network [19] on unlabeled data in order to further add them to the training sample [17]. In this case, the speed of classification of MSM’s data depends on the speed of self-learning of the Kohonen network, after which it is necessary to solve the problem of matching existing weakly labeled data with the obtained cluster markers. In addition, adjusting the weights of neurons according to the WTA algorithm with gradient optimization in the process of learning the Kohonen network in a training sample consisting of conflicting weakly labeled data requires significant time, which is unacceptable. Therefore, the use of genetic algorithms with the inherent possibility of random mutation to adjust the weights of neural networks reduces the negative impact of erroneous decisions on the overall level of learning speed and reliability of classification [17, 18].

Thus, it is important to solve the scientific and practical problem of developing models for presenting weakly structured and weakly labeled data heterogeneous MSM’s data and methods of their classification based on Kohonen network training using genetic algorithms to create IS MSM. The practical use of the developed IS MSM will allow to make reliable classification decisions about the lifestyle or working conditions of each of the respondents of the studied target audience

2 Review of Literature

Research shows that the decision-making process in IS MSM consists of four stages: formation of research hypotheses about the composition of the target audience (social environment), preparation and storage of MSM data, making

classification decisions to confirm or refute research hypotheses, and interpretation/evaluation of results classification (Fig. 1) [4]. The intellectual activity of a sociologist expert on the formation of research hypotheses and classification decisions on the state of public health of the target audience based on the analysis of quantitative metrics or qualitative assessments of sociodemographic, medical, psychophysical, behavioral, geographical or any other characteristics generalization of data consisting of self-assessments of the population of their health, quality of life, satisfaction with medical and social services.

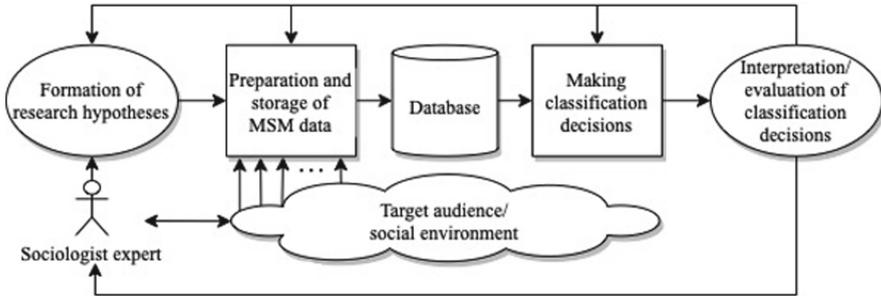


Fig. 1. General structure of a typical IS MSM

Analysis of MSM’s data shows that they are heterogeneous because they are periodically updated from completed forms of surveys, questionnaires and interviews or e-mails, business cards, pdf and txt-files, instant messages in different messengers (WhatsApp, Viber, Facebook Messenger, Skype, ICQ, Google Hangouts, etc.), documents, web pages, invoices, audio/video, checks and contracts, pictures, etc. In addition, heterogeneous MSM input data are weakly structured (WS) because they are collected from different sources, interpreted using different and not always related scales, and often contradict each other [21]. It is known that the analysis of large arrays of weekly structured heterogeneous data requires long-term manual work of qualified sociologists, so for MSM inputs, which are periodically updated, such a classification decision-making process is in some cases impossible [20].

On the other hand, research shows that the creation of IS MSM based on machine learning with a teacher (supervised learning) allows to automate the process of obtaining classification decisions about the target audience only in the presence of labeled empirical quantitative or categorical data [3,23]. However, it was previously established that it is impossible to obtain a sufficient amount of such labeled data (training sample) in the case of classification of weekly structured heterogeneous MSM data. Therefore, in practice, heterogeneous MSM’s data are weakly labeled (WL) and semi-supervised learning (SSL) methods are used to obtain classification solutions when processing such unlabeled data [8]. Research shows that one of the common ways to implement SSL methods is to use the self-learning neural network of the Kohonen neural network

with a competing layer on unlabeled data in order to further add them to the training sample [14]. But in this case, the speed of classification of MSM's data depends on the speed of self-learning of the Kohonen network [19], after which it is necessary to solve the problem of matching existing pre-labeled data with the obtained cluster markers. In addition, the situation with the required speed of classification of MSM's data is complicated by the fact that adjusting the weights of neurons according to the WTA algorithm with gradient optimization in the process of learning the Kohonen network in a training sample consisting of conflicting WL data requires considerable time.

On the other hand, it is shown that in the case of large areas of permissible values of neural network parameters, and the learning error surface has a complex relief, the use of genetic algorithms (GA) for learning neural networks with their inherent possibility of random mutation reduces the negative impact of erroneous decisions on the overall level of reliability of the classification [16]. Considering the need to solve the problem of classification of weekly structured and weekly labeled (WS & WL) heterogeneous data in the IS MSM, the purpose and objectives of the study are formulated

3 Problem Statement

The goal of the research is to reduce the time for making classification decisions without losing the level of their reliability in the IS MSM by developing models for the presentation of WS & WL heterogeneous data and methods for their classification based on the Kohonen network. To achieve this goal, the following tasks were solved:

- the methods of classification of heterogeneous data in the existing IS MSM are analysed and the directions of research are chosen;
- models for presenting weekly structured and weekly labeled heterogeneous MSM data have been developed;
- developed methods for classification of weekly structured and weekly labeled heterogeneous MSM data;
- MSM IS was developed and tested on MSM practical data.

4 Materials and Methods

4.1 Development of Models Presentation of WS & WL of Heterogeneous MSM Data

The task of classifying of respondents of the target audience in the IS MSM is formalized as follows. Let's consider that Target Audience consists of many *Respondents* R_i :

$$TA = \{R_1, R_2, \dots, R_i, \dots, R_n\}, j = \overline{1, n}, \quad (1)$$

where: R_i is the survey respondent, n is the number of respondents.

According to the structure of a typical IS MSM (Fig. 1), the sociologist hypothesizes that each respondent R_i of the target audience TA belongs to the class t^k : from the set of class values:

$$T = \{t^1, \dots, t^k, \dots, t^p\}, k = \overline{1, p}, \tag{2}$$

where: p is the power of the set of classes $|T|$. Thus $p \ll n$.

On the other hand, for making classification decisions regarding the respondents in the IS (Fig. 1) use WS sets of heterogeneous MSM's data, the value, type and format of which depends on the method of obtaining input information. Respondent data extracted from various sources of information are *detailed data*. Then, taking into account representation (1), it is proposed to set the detailed data D_{R_i} in the m -dimensional space of properties using the following model:

$$D_{R_i} = \langle \{V_\phi\}_j, T_j, F_j, S_j, Q_j, Mt_j \rangle, j = \overline{1, q}, \tag{3}$$

where: $\{V_\phi\}_j, T_j, F_j, S_j, Q_j, Mt_j$ is is set of discrete (qualitative or quantitative) values, type, format, source, quality assessment, procedure of aggregation/transformation of data j -property; q is the number of properties of the respondent R_i . Thus $q \ll n$.

To reduce the time to make a classification decision t^k without losing the level of reliability, the set of discrete values $\{V_\phi\}_j$ detailed data D_{R_i} of each type taking into account the presentation format using the procedure Mt_j is aggregated and transformed into feature space:

$$x_i = Mt_j(\{V_\phi\}_j), \tag{4}$$

where: the procedure Mt_j depending on the type, format, source and quality assessment of the obtained discrete values $\{V_\phi\}_j$ is selected from many possible functions of quantization (Quant), sorting (Sort), merge (Merg), grouping (Group), data setup (DSett), tabular value substitution (TSubst), value calculation (CalcVal), data encoding (DEncod), normalization (scaling) (Norm (Scal)), etc. Thus:

$$Mt_j \in \{Quant, Sort, Merg, Group, DSett, TSubst, CalcVal, DEncod, Norm(Scal)\} \tag{5}$$

Then the model of aggregated data A_{R_i} of the respondent R_i (1) in the q -dimensional space of features taking into account representations (2)–(4) is given by means of a tuple:

$$A_{R_i} = \{x_1, \dots, x_i, \dots, x_q, t_i^k\}, j = \overline{1, q}, k = \overline{1, p}, \tag{6}$$

where: q is the dimension of feature space, t_i^k is a marker of class k from the set (2), to which the i -respondent belongs.

The marker of class is determined in manual or automated mode in three ways:

- at the discretion of the respondent R_i only in cases of surveys, questionnaires or interviews is a manual mode;
- as a result of the analysis of values of a set of aggregated data $A_{R_i} = \{x_j\}$ which is carried out by the expert sociologist on the basis of own experience and knowledge is a manual mode;
- as a result of construction of the classification rule $a : X \rightarrow T$ by means of machine learning with the teacher as which the training sample $A_R = \{A_{R_i}\}$ is the automated mode.

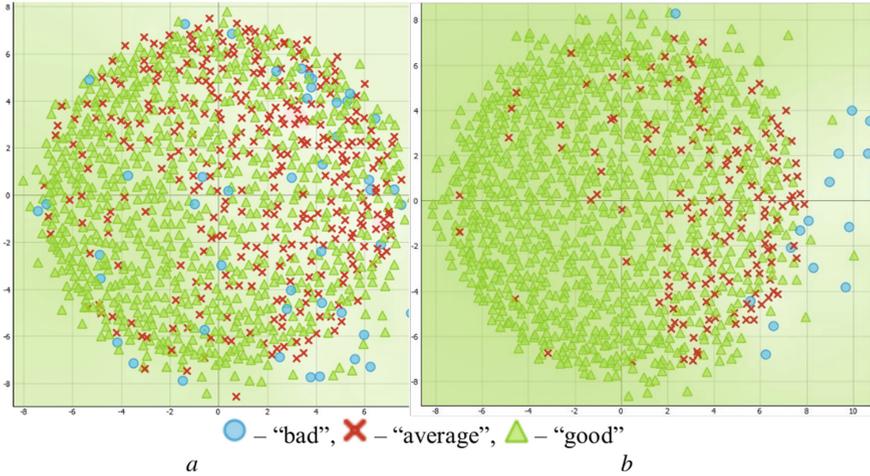


Fig. 2. Map of similarities of MSM respondents by lifestyle for class markers $q = 3$ a) by respondent’s own discretion; b) by decision of a sociologist

Figure 2 shows a scatter plot in the form of a similarity map of MSM respondents “Ukraine - lifestyle” [1], which is based on a *three-stage scenario*. At the first stage, according to 114 properties for 1143 respondents, the detailed input data D_R , which characterize the lifestyle of *three classes* [10], were removed. In the second stage, the data D_R according to (3)–(4) are aggregated and transformed into a space of 23 features, thus obtaining $A_{R_i} = \{x_j\}$, $j = \overline{1, 23}, k = \overline{1, 3}$. In the third stage A_R , using multidimensional scaling (scaling), transformed into two-dimensional space so-called *Multidimensional scaling - MDS* projections [11], which are displayed taking into account the markers of classes t_i^k , obtained manually by the respondent (Fig. 2a), or an expert (Fig. 2b).

Therefore, according to (1) and (6) in matrix form, the *training set* of MSM’s data is a set of aggregated data of respondents R_i of the target audience TA and consists of labeled and unlabeled data [21]:

$$A_R^{train} = A_R^l \cup A_R^u, \tag{7}$$

where:

$$\begin{aligned}
 A_R^l = \langle X, T \rangle &= \left\langle \begin{bmatrix} x_{11} & \dots & x_{1q} \\ \dots & \dots & \dots \\ x_{l1} & \dots & x_{lq} \end{bmatrix}, \begin{bmatrix} t_1^1 \\ \dots \\ t_l^p \end{bmatrix} \right\rangle, \\
 A_R^u = \langle X \rangle &= \left\langle \begin{bmatrix} x_{l+11} & \dots & x_{l+1q} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nq} \end{bmatrix} \right\rangle.
 \end{aligned}
 \tag{8}$$

The matrix X is the set of features of the respondents R_i in the q -dimensional space of features, and the vector T is a finite set of numbers (names, labels, markers) of classes (2).

The test set has the form:

$$A_R^{test} = \left\langle \begin{bmatrix} x_{n+11} & \dots & x_{n+1q} \\ \dots & \dots & \dots \\ x_{g1} & \dots & x_{gq} \end{bmatrix} \right\rangle.
 \tag{9}$$

Then the problem of classification of MSM's data in IS, in terms of machine learning with partial involvement of the teacher, is proposed to be formalized as follows: there is an unknown *target dependence* - $a : X \rightarrow T$, the values of which are known only for a finite number of respondents R_i from the marked part of the training sample A_R^l . Taking into account the data from the unmarked part of the training sample A_R^u , it is necessary to build an algorithm $a^* : X \rightarrow T$, capable of classifying any respondent $R_i \in A_R^{test}$ with a given reliability. To classify MSM's data with partial teacher involvement, it is proposed to first conduct a cluster analysis using the Kohonen neural network, which solves the problem of dividing the training sample A_R^{train} MSM's data into disparate clusters:

$$Z = \{ z^1, \dots, z^m, \dots, z^p \}, m = \overline{1, p},
 \tag{10}$$

where: z^m is a cluster marker containing vectors that are similar in characteristics to the vector A_{R_i} (6) from the training sample A_R^{train} , $i = \overline{1, n}$; n is the number of vectors in the sample [15]:

$$z^m = \left\{ A_{R_i}, A_{R_j} \mid \begin{matrix} A_{R_i} \in A_R^{train}, A_{R_j} \in A_R^{train} \\ d(A_{R_i}, A_{R_j}) < \sigma \end{matrix} \right\},
 \tag{11}$$

where: $d(A_{R_i}, A_{R_j})$ is a measure of closeness between respondents, called distance; σ is a value that determines the proximity threshold for the inclusion of respondents in one cluster.

The resulting set of clustered data A_R^{train*} looks like:

$$A_R^{train*} = \langle X, Z \rangle = \left\langle \begin{bmatrix} x_{11} & \dots & x_{1q} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nq} \end{bmatrix}, \begin{bmatrix} z_1^1 \\ \dots \\ z_n^p \end{bmatrix} \right\rangle,
 \tag{12}$$

4.2 Development of a Method for Matching Class Markers and Clusters in the Teaching of the Kohonen Network with a Partial Involvement of the Teacher

It is proposed [6] to automate the learning process of the Kohonen network using WM heterogeneous data MSM A_R^{train} (7) using a method that includes five stages (Fig. 3).

Stage 1. In accordance with the size of the feature space q and the power of the set of classes p , a competing layer of the Kohonen network is created and taking into account the values of A_R^{train} (7) the matrix of weights $\{w_{ij}\}$ is initialized $\{b_j\}$.

Stage 2. The procedure of self-organization of the competing layer is implemented on the basis of the classical iterative algorithm WTA. The values of the vectors $\{x_{ki}\}$ from the training sample A_R^{train} are successively fed to the input of the competing layer. The winner’s neuron is determined by minimizing distances: $d(x_{ki}, w_{ij}) \rightarrow \min d(x_{ki}, w_{ij})$.

The Euclidean distance $d(x_{ki}, w_{ij}) = \sqrt{\sum_{i=1}^q (x_{ki} - w_{ij})^2}$ between the elements of the input vectors x_{ki} and the weights w_{ij} is used as a measure of distance.

Stage 3. To obtain the cluster markers at the input of the Kohonen network obtained in stage 2 by the matrix W and the vector B from the training sample A_R^{train} , the values of the vectors $\{x_{ki}\}, i = \overline{1, q}$, q is the dimension of space signs. The network response is formed in the form of a binary vector $\{y_{kj}\}, j = \overline{1, p}$, where p is the number of neurons in the competing layer for the k -th vector. In this case, $\{y_{kj}\} = 1$, if j is the number of the winning neuron and $\{y_{kj}\} = 0$ in all other cases.

The result of the stage is the initial vector $\{z_{kj}\}$, the value of the k -th element of which is assigned the ordinal number of the winning neuron of the j -th binary vector $\{y_{kj}\}$.

Thus we obtain a preliminary solution as follows:

$$R^* = \langle X, Y, Z \rangle = \left\langle \begin{bmatrix} x_{11} & \dots & x_{1q} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nq} \end{bmatrix}, \begin{bmatrix} y_{11} & \dots & y_{1p} \\ \dots & \dots & \dots \\ y_{n1} & \dots & y_{np} \end{bmatrix}, \begin{bmatrix} z_1^1 \\ \dots \\ z_n^p \end{bmatrix} \right\rangle, \quad (13)$$

where: X is a features set of the respondents R_i , Y is a binary matrix of responses of the competing layer, Z is a set of cluster markers.

Stage 4. To reconcile the existing class markers of the set T (2) to which the respondent R_i belongs from the marked data A_R^l (8) with the values of the marker clusters of the set Z (13), a corresponding method was developed that requires 4 steps (Fig. 4) [12, 24].

Step 4.1. For all examples i of the marked part A_R^l of the training sample formation a two-dimensional histogram D (square matrix $\{D_{mk}\}$) of pairwise coincidences of values of existing markers of classes T and the received markers of clusters Z , $D_{mk} = D_{mk} + 1$, if $t_i^k = z_i^m$, where: $m, k = \overline{1, p}$.

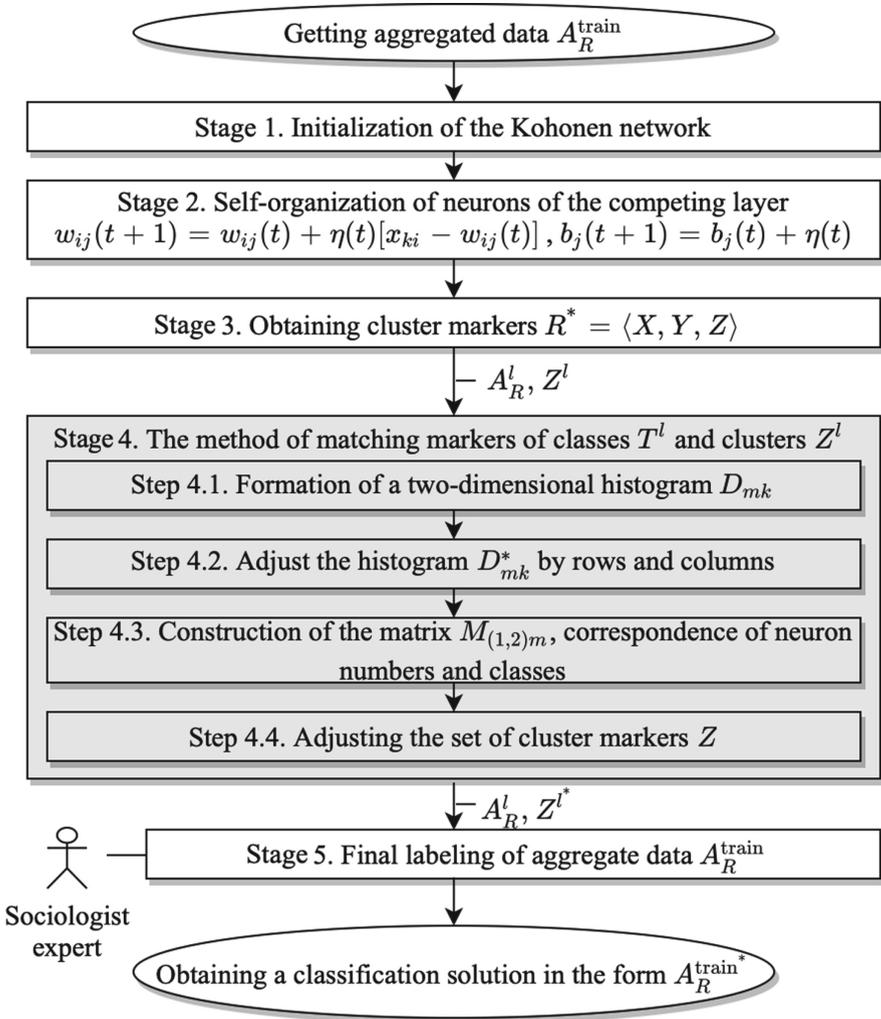


Fig. 3. Stages of Kohonen network training method with the partial involvement of a teacher General structure of a typical IS MSM

Step 4.2. In loops (rows and columns), reset the elements of the histogram D_{mk} , leaving unchanged the values of those that satisfy the rule:

$$D_{mk}^* = \begin{cases} 0 \\ D_{mk}, \langle \max_m(D_{mk}) = \max_k(D_{mk}) | m = k | \max(D_{mk}) \ll 0 \rangle \end{cases} \quad (14)$$

Adjustment of the histogram D_{mk} is performed until only one non-zero value remains in each row and column.

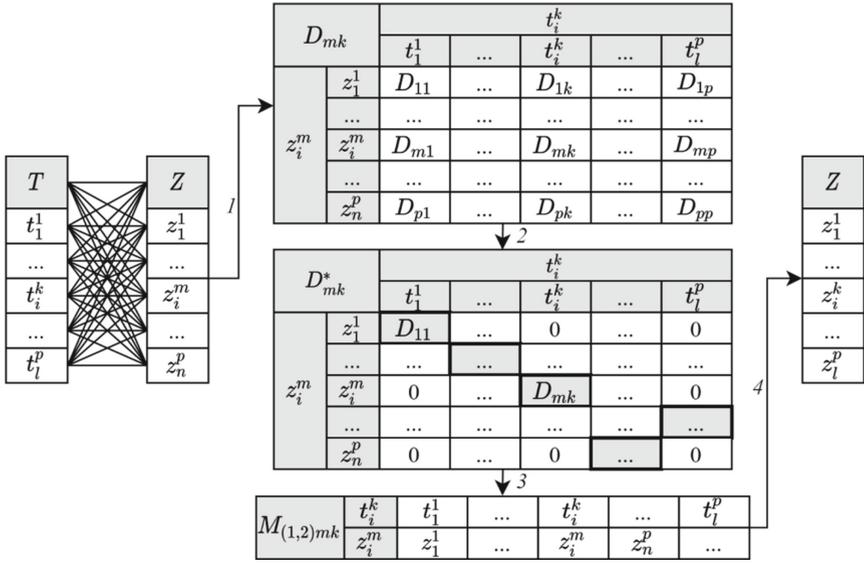


Fig. 4. Scheme of steps of the method of matching class markers and clusters

Step 4.3. Converting the adjusted histogram D_{mk}^* into a matrix of correspondence of values of markers of classes and clusters $M_{(1,2),m}$ for all examples of the training sample A_R^l .

Step 4.4. Correct the vector of markers of clusters Z (3) using the matrix of correspondences $M_{(1,2),m}$, form a classification solution for the marked part of the training sample in the form:

$$A_R^{l*} = \langle X, T, Z \rangle = \left\langle \begin{bmatrix} x_{11} & \dots & x_{1q} \\ \dots & \dots & \dots \\ x_{l1} & \dots & x_{lq} \end{bmatrix}, \begin{bmatrix} t_1^1 \\ \dots \\ t_l^p \end{bmatrix}, \begin{bmatrix} z_1^1 \\ \dots \\ z_l^p \end{bmatrix} \right\rangle \quad (15)$$

and pass it on to a sociologist for further evaluation and interpretation.

Stage 5. The final marking of the aggregated data A_R^{train} is performed by a sociologist for the marked part A_R^l and the unmarked part A_R^u based on the solution (15). As a rule, the training sample A_R^l is not balanced, so the reliability of the classification is determined using the metrics *True Positive Rate - TPR*, which shows the percentage of positive class examples found by the classifier, and *False Positive Rate - FPR*, which shows the percentage of negative class examples erroneously assigned by the classifier to positive class [20]:

$$TPR = \frac{TP}{TP + FN} \times 100\%, \quad (16)$$

$$FPR = \frac{FP}{FP + TN} \times 100\%, \quad (17)$$

where: TP are correctly classified positive examples; FN are positive examples classified as negative (type II error), FP are negative decisions classified as positive (type I error); TN are true negative examples, classified as negative

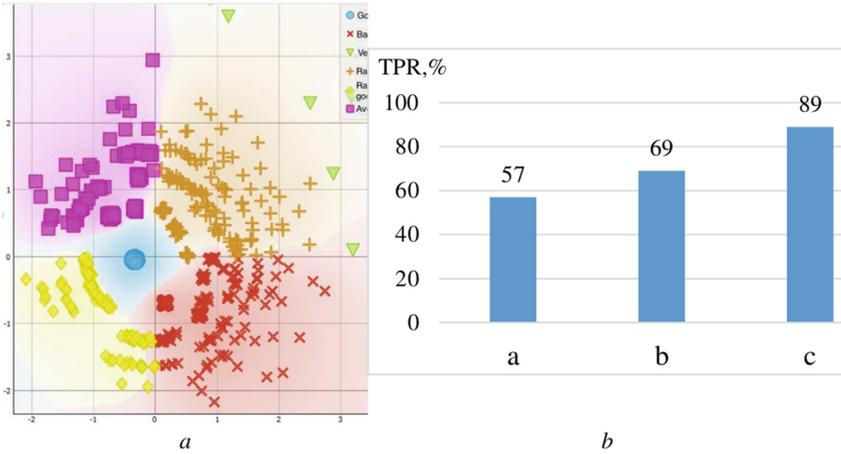


Fig. 5. Results of the implementation of the method of teaching the Kohonen network with the partial involvement of teachers of MSM respondents by lifestyle: a) similarity map; b) the reliability of the classification

Figure 5, a shows a similarity map of MSM respondents “Ukraine - lifestyle”, which is built on the same scenario as the maps shown in Fig. 2a and 2b, but for six classes $\{t^k\}$, $k = \overline{1,6}$ in the marked part of the sample, namely: \circ - “good”, \times - “bad”, ∇ - “very bad”, $+$ - “rather bad”, \diamond - “very good”, \square - “average”.

The markers of clusters z^k in the unmarked part of the sample A_R^u were obtained in an automated mode based on the self-organization of the Kohonen network (stage 2, see Fig. 3) and coordinated with existing markers of classes of the marked part of the sample A_R^l according to the developed method (stage 4, see Fig. 3). Figure 5b shows the results of calculations of the reliability of the classification by metric (16) (columns a, b, c) for aggregate data A_R , MDS projection, which are shown in Fig. 2a, Fig. 2b and Fig. 5a, respectively. It should be noted that if the time of making classification decisions by the respondent’s own decision to take as a relative “unit” (column a), then in automated mode (column c) time is *reduced by half* compared to the manual labor of a sociologist (column b).

Thus, the use of the method of matching class markers and clusters in the process of learning the Kohonen network according to the WTA algorithm with gradient optimization allowed to automate the classification decisions of WS & WL heterogeneous MSM data. But the reliability of the classification in the case of close and unbalanced classes is less than 90%. Therefore, to increase it, a

heuristic adjustment of weights in the learning process of the Kohonen network is proposed.

4.3 Development of a Method for Classification of SS and CM Heterogeneous MSM Data

By matching the existing class markers and the obtained cluster markers to further assess the suitability of chromosomes in the heuristic adjustment of weights in the learning process of the Kohonen network, a method of classification of WS & WL heterogeneous MSM data was developed, consisting of six stages (Fig. 6):

Stage 1. Initialization of the Kohonen network and formation of the initial population. The initialization of the Kohonen network follows the description of *Stage 1* (Fig. 3). The formation of the initial population Pl is to create chromosomes in the amount of N_p . Each chromosome CH_h is a vector of random real numbers, which is correlated with individual weights and displacements: $CH_h = \langle W, B \rangle^h, h = \overline{1, N_p}$:

$$CH_h = \{w_{11}^h, \dots, w_{1q}^h, \dots, w_{p1}^h, \dots, w_{pq}^h, b_1^h, \dots, b_p^h\}. \tag{18}$$

If we represent CH_h as a set of genes $\{g_1, \dots, g_G\}$, where $G = p \times q + p$ is the power of CH_h , then the population Pl is as follows:

$$Pl = \begin{bmatrix} CH_1 \\ \dots \\ CH_{N_p} \end{bmatrix}, \begin{bmatrix} g_{11} & \dots & g_{1G} \\ \dots & \dots & \dots \\ g_{N_p 1} & \dots & g_{N_p G} \end{bmatrix}. \tag{19}$$

Stage 2. Obtaining cluster markers $R^ = \langle Pl, \{Y^h\}, \{Z^h\} \rangle$.* To obtain preliminary solutions in the form of cluster markers at the entrance of the competing layer, which is a population Pl (19), the values of the vectors $\{x_{ki}\}$ from the training sample A_R^{train} train are sequentially given and the Euclidean distance for each chromosome CH_h is calculated network response in the form of a binary vector $\{y_{kj}\}^h, k = \overline{1, n}, j = \overline{1, p}, h = \overline{1, N_p}$. The result of the step is the initial vector $\{y_{kj}\}^h$, the value of the k -th element of which is assigned the ordinal number of the winning neuron j of the binary vector $\{y_{kj}\}^h$. Thus, for the population Pl taking into account each chromosome $CH_h, h = \overline{1, N_p}$ we obtain the set of previous solutions as follows:

$$R^* = \langle Pl, \{Y^h\}, \{Z^h\} \rangle = \left\langle \begin{bmatrix} x_{11} & \dots & x_{1q} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nq} \end{bmatrix}, \begin{bmatrix} y_{11} & \dots & y_{1p} \\ \dots & \dots & \dots \\ y_{n1} & \dots & y_{np} \end{bmatrix}^h, \begin{bmatrix} z_{11} \\ \dots \\ z_{np} \end{bmatrix}^h \right\rangle. \tag{20}$$

where: X is the set of characteristics of the respondents R_i , Y^h is the binary matrix of responses of the competing layer, Z^h is the set of cluster markers.

Stage 3. Matching the markers of classes T and clusters Z^h is performed for each chromosome $CH_h, h = \overline{1, N_p}$, according to the developed method (Fig. 4), we obtain:

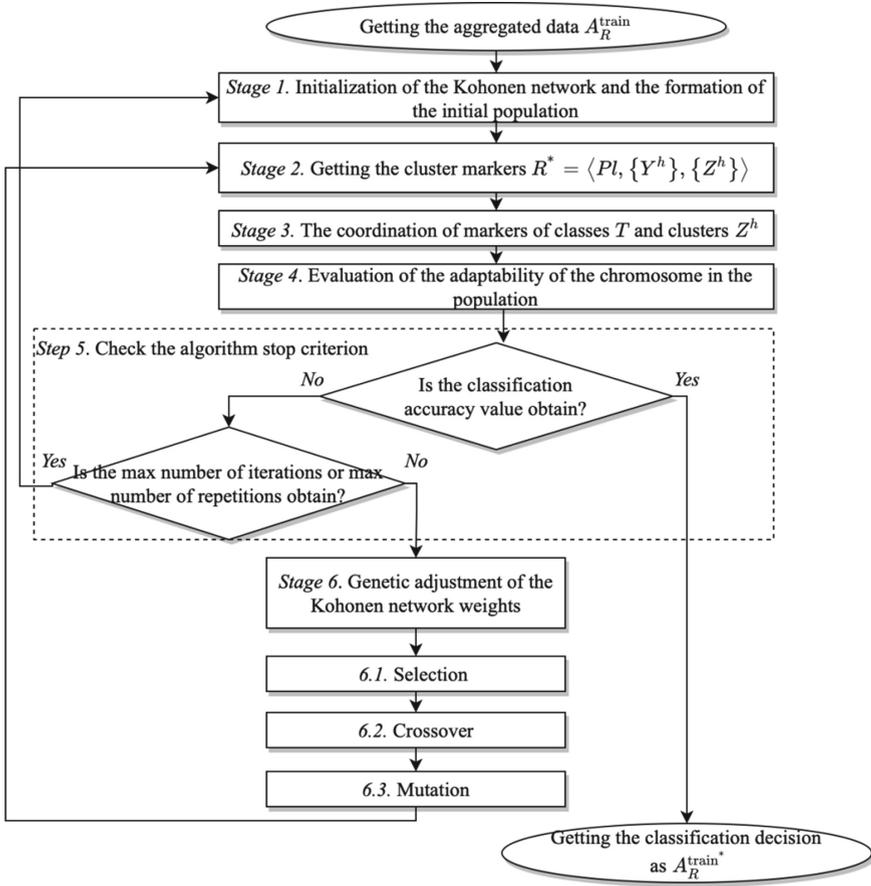


Fig. 6. Scheme of the method of classification of WS & WL heterogeneous MSM data

$$R^{**} = \langle Pl, X, T, \{Z^h\} \rangle = \left\langle \begin{bmatrix} x_{11} & \dots & x_{1q} \\ \dots & \dots & \dots \\ x_{l1} & \dots & x_{lq} \end{bmatrix}, \begin{bmatrix} t_1^1 \\ \dots \\ t_l^p \end{bmatrix}, \begin{bmatrix} z_1^1 \\ \dots \\ z_l^p \end{bmatrix}^h \right\rangle. \quad (21)$$

Stage 4. Assessment of chromosome fitness in the population. In the cycle, based on the existing markers of classes T and the agreed markers of clusters Z^h , we calculate the reliability of the classification TPR^h by metric (16) for each chromosome CH_h .

Stage 5. Check the stop criterion of the algorithm. Criteria for stopping the algorithm are: successful completion - achieving the required level of classification reliability; unsuccessful completion - exceeding the maximum number of iterations or repetitions of the value of the reliability of the classification, which

means the degeneration of the population and requires the transition to reinitialization of the Kohonen network. If the conditions of the stop are not met, we move on to the sixth stage.

Stage 6. Heuristic adjustment of Kohonen network weights is performed by classical procedures of selection (selection), crossing (crossover) and mutation (mutation) of chromosomes according to the assessment of their suitability obtained in Stage 5. To perform the selection, we sort the populations in descending order of the assessment of fitness TPR^h . From the sorted sequence we choose the first two (CH_1^w (winner) and CH_2^{vw} (vice winner)) and the last two ($CH_{N_p-1}^{vl}$ (vice loser) and ($CH_{N_p}^l$ (loser)) chromosomes. The winners are crossed to obtain two “adapted” chromosomes in order to further replace the “loser” chromosomes.

For crossbreeding, determine the random point of crossover, which divides the chromosome into two parts and exchange chromosome genes (Fig. 7).

The mutation operation is helpful in creating a new chromosome and is performed by adding to each gene of the chromosome CH_h a random number distributed according to a uniform law:

$$CH_h^{new} = CH_h + rand(-0,05max(CH_h), +0,05max(CH_h)). \quad (22)$$

A new population Pl^{new} is formed from the new chromosomes, which is passed as input data to perform the actions of stage 2.

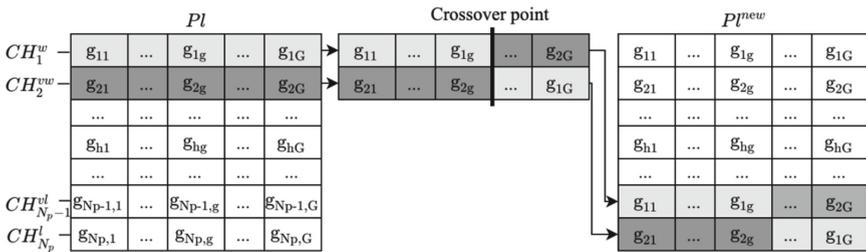


Fig. 7. Example of a crossover operator

The results of testing the method of heuristic adjustment of weights in the study of the Kohonen network for the example of classification of data on “Fisher’s irises” [13, 17] with classes 2 and 3, which intersect, are shown in Fig. 8.

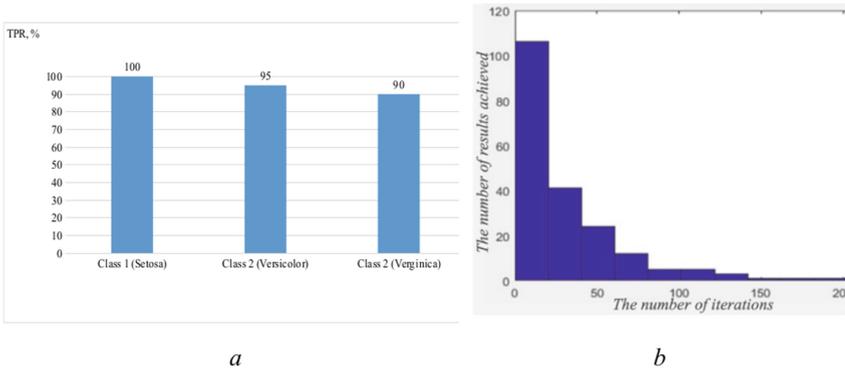


Fig. 8. The result of testing the method of heuristic adjustment of the weights of the Kohonen network: a) TPR, b) training diagram

The analysis of the above learning chart shows that all 200 Kohonen network learning trials were successfully completed with a classification accuracy of more than 95% and 90% for intersecting classes. At the same time, in most of the subjects (176) less than 50 epochs were required to study the Kohonen network (Fig. 8, b), which is faster than according to the WTA algorithm with gradient optimization.

Data on approbation of the developed method of WS & WL classification of heterogeneous data of MSM respondents “Ukraine - lifestyle” (see Fig. 2 and Fig. 8) are given in Table 1. Comparative results on time and reliability of classification decisions based on Kohonen network with weighting WTA algorithm and *heuristic adjustment* (HA) highlight the advantage of an improved method of classification on *training* and *test* samples of heterogeneous data of MSM respondents.

Table 1. Comparison of the results of the classification of SS & CM heterogeneous MSM data

Indicators	WTA		HA	
	Training	Testing	Training	Testing
Reliability of classification (TPR), %	88,96%	81,08%	94,99%	86,06%
Work time, s	1541		1217	

To test the developed models and methods, IS MSM was created (Fig. 9) [6].

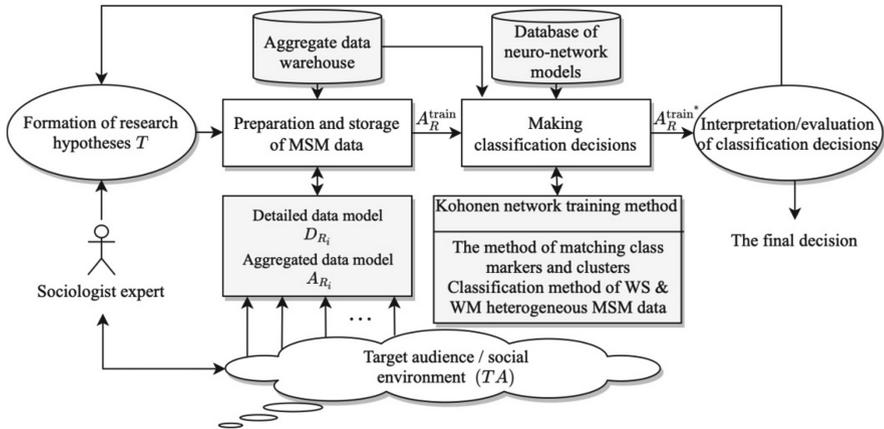


Fig. 9. Structure of IS MSM, taking into account the proposed models and methods.

The subsystem of preparation of initial data of IS MSM is built on the principle of ETL-systems containing means:

- *Extraction* of detailed data of different formats from different types of information sources about the object of management and their storage, taking into account the developed model of detailed data;
- *Aggregation* and *transformation* of MSM’s data into a single format based on the developed model of aggregated data;
- *Loading* aggregate data (training samples) into the repository for further use as a training sample.

The implementation of the Kohonen network training unit takes into account the developed method of classification of WS & WL heterogeneous MSM’s data by matching existing class markers and obtained cluster markers for further assessment of chromosome fitness in heuristic weight adjustment in Kohonen network learning process. The obtained matrices of weights are loaded into the *database of neural network models* and then together with aggregated data are used to clarify the research hypothesis about the composition of the target audience, support classification decisions to confirm or refute the refined research hypothesis.

The developed IS MSM is used to classify the data “working conditions” of employees of the company “Service-Grain” in a comprehensive examination of the factors that affect the health of the target audience. Working conditions depend on the quantitative and qualitative characteristics that determine the levels of aerosol, electromagnetic, acoustic, chemical and biological effects, ionizing radiation, microclimate, lighting and vibration. As a result of analysis of MSM’s data according to the proposed scale, which depends on the level of maximum allowable concentration of the studied factor, the sociologist evaluates the state of working conditions as: “optimal”, “acceptable”, “harmful”, “dangerous”,

“extreme”. The training sample consisted of 1,200 examples. In the course of the research, the expert sociologist has the opportunity to compare and correct his decision in accordance with the automated solution of IS MSM. The reliability of the classification was determined by metrics (16) - *TPR* and (17) - *FPR*.

Analysis of the calculations of error values before and after the adjustment shows an increase in the relative share of *TPR* by an average of 20% and a decrease of 50% *FPR* for all groups of working conditions. At the same time, after adjusting the decisions of the sociologist, *TPR* estimates are more evenly distributed for all studied levels of influence from the group of working conditions (Fig. 10a). A decrease in the relative proportion of true negative cases (*FPR*) is especially noticeable for the levels of chemical and biological effects (Fig. 10b).

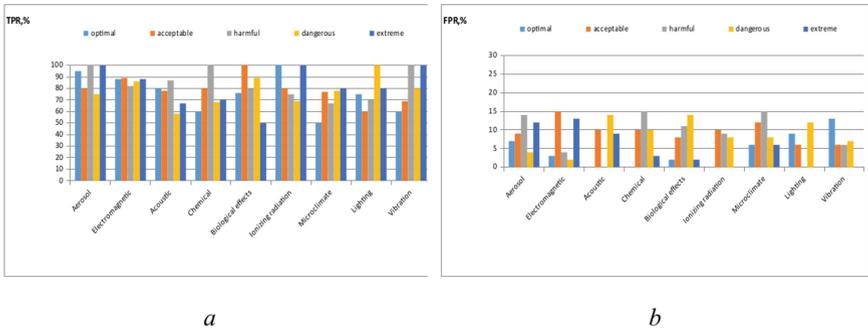


Fig. 10. The results of the classification of working conditions: a) *TPR*; b) *FPR*

5 Conclusions and Future Work

Thus, the paper creates and studies models of WS & WL presentation of heterogeneous MSM’s data and methods of their classification, in order to reduce the time for making classification decisions without losing the level of their reliability in the IS MSM. The main results of the work are the following:

- To improve the preprocessing subsystem in the IS MSM, models of representation of WS & WL heterogeneous MSM’s data in property and feature spaces are developed, taking into account values, types, formats, sources, quality estimates and procedures for aggregation/transformation of detailed data properties. This allowed to formalize the creation of training and test samples for further classification with the partial involvement of teachers.
- To obtain additional class markers in the unmarked part of the study sample, a method of matching class and cluster markers in Kohonen network training was developed. Approbation of the method showed that the time of classification decisions in the automated mode was reduced by half compared to the manual mode.

- A method of heuristic adjustment of weights in the process of learning the Kohonen neural network with the coordination of class markers and clusters to calculate the assessment of chromosome fitness in the population of the genetic algorithm for each example from the training sample. An analysis of the above Kohonen network learning diagrams for a series of tests in a sample of intersecting classes shows that the classification is more than 90–95% reliable in less than 50 learning epochs, which is 50% more than the WTA gradient optimization algorithm.
- A classification method based on the harmonization of existing class markers and the obtained cluster markers for further assessment of chromosome fitness in heuristic adjustment of weights in the process of learning the Kohonen network has been developed. Comparative results on the time and reliability of classification decisions based on the Kohonen network with adjustment of weights according to the WTA algorithm and heuristic adjustment highlight the advantages of an improved method of classification in training and test samples of heterogeneous data of MSM respondents “Ukraine - lifestyle” increase by 5% and time reduction by 21%.
- Practical use of WS & WL models of heterogeneous MSM’s data and methods of their classification allowed to develop IS MSM, which includes updated subsystems of preprocessing and classification decision-making. Approbation of IS MSM on data on the lifestyle and working conditions of respondents showed an increase in reliability and a decrease in time positive, which is confirmed by the relevant acts of implementation.

References

1. Sociological survey data “Ukraine - life style”. <http://edukacijainauka.pl/limesurvey/index.php/lang-pl>
2. Alzubaidi, L., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **8**(53) (2021). <https://doi.org/10.1186/s40537-021-00444-8>
3. Arsirii, O., Antoshchuk, S., Babilunha, O., Manikaeva, O., Nikolenko, A.: Intellectual information technology of analysis of weakly-structured multi-dimensional data of sociological research. *Adv. Intell. Syst. Comput.* **1020**, 242–258 (2020). https://doi.org/10.1007/978-3-030-26474-1_18
4. Arsirii, O., Babilunha, O., Manikaeva, O., Rudenko, O.: Automation of the preparation process weakly-structured multi-dimensional data of sociological surveys in the data mining system. *Herald Adv. Inf. Technol.* **1**(1), 9–17 (2018). <https://doi.org/10.15276/hait.01.2018.1>
5. Arsirii, O., Manikaeva, O.: Models and methods of intellectual analysis for medical-sociological monitoring data based on the neural network with a competitive layer. *Appl. Aspects Inf. Technol.* **2**(3), 173–185 (2019). <https://doi.org/10.15276/aait.03.2019.1>
6. Manikaeva, O., Arsirii, E., Vasilevska, A.: Development of the decision support subsystem in the systems of neural network pattern recognition by statistical information. *East. Eur. J. Enterp. Technol.* **6**(4(78)), 4–12 (2015). <https://doi.org/10.15587/1729-4061.2015.56429>

7. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: a brief survey. *IEEE Signal Process. Mag.* **34**, 26–38 (2017). <https://doi.org/10.1109/MSP.2017.2743240>
8. Barsegyan, A.A., Kupriyanov, M.S., Stepanenko, V.V., Holod, I.I.: *Metody i modeli analiza dannyih: OLAP i Data Mining*. BHV-Peterburg, SPb (2004)
9. Belciug, S., Gorunescu, F.: Era of intelligent systems in healthcare. In: *Intelligent Decision Support Systems—A Journey to Smarter Healthcare*. ISRL, vol. 157, pp. 1–55. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-14354-1_1
10. Buja, A., Swayne, D.F., Littman, M.L., Dean, N., Hofmann, H., Chen, L.: Data visualization with multidimensional scaling. *J. Comput. Graph. Stat.* **17**(2), 444–472 (2008). <https://doi.org/10.1198/106186008X318440>
11. Chapman, P., et al.: *Crisp-dm 1.0: step-by-step data mining guide* (2000)
12. Corbetta, P.: *Social Research: Theory, Methods and Techniques*. Sage, London (2011). <https://doi.org/10.4135/9781849209922>
13. Djouzi, K., Beghdad-Bey, K.: A review of clustering algorithms for big data. In: *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pp. 1–6 (2019). <https://doi.org/10.1109/ICNAS.2019.8807822>
14. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in data bases. *AI Mag.* **17**(3), 37–54 (1996). <https://doi.org/10.1609/aimag.v17i3.1230>
15. Härdle, W., Simar, L.: *Applied Multivariate Statistical Analysis* Free preview. Springer, New York (2012). <https://doi.org/10.1007/978-3-030-26006-4>
16. Hartigan, J., Wong, M.: Algorithm AS 13: a k-means clustering algorithm. *Appl. Stat.* **28**(1), 100–108 (1979). <https://doi.org/10.2307/2346830>
17. Haykin, S.: *Neural Networks and Learning Machines*, 3rd edn. McMaster University, Ontario (2009)
18. Katoch, S., Chauhan, S., Kumar, V.: A review on genetic algorithm: past, present, and future. *Multimedia Tools Appl.* **80**(5), 8091–8126 (2021). <https://doi.org/10.1007/s11042-020-10139-6>
19. Kohonen, T.: The self-organizing map. *Proc. IEEE* **78**(9), 1464–1480 (1990). <https://doi.org/10.1109/5.58325>
20. Merkert, J., Müller, M., Hubl, M.: A survey of the application of machine learning in decision support (2015). <https://doi.org/10.18151/7217429>
21. Praveen, S., Chandra, U.: Influence of structured, semi-structured, unstructured data on various data models. *Int. J. Sci. Eng. Res.* **8**(12), 67–69 (2017)
22. Qin, L., et al.: Prediction of number of cases of 2019 novel coronavirus (Covid-19) using social media search index. *Int. J. Environ. Res. Public Health* **17**(7) (2020). <https://doi.org/10.3390/ijerph17072365>
23. Rudenko, A.I., Arsirii, E.A.: Metodika intelektualnogo analiza slabostrukturirovannyih mnogomernyih dannyih sotsiologicheskikh oprosov [Methods of intellectual analysis of poorly structured multidimensional data of sociological surveys]. In: *Modern Information Technology 2018*, pp. 168–169. MON Ukrayini, Odes. Nats. politeh. un-t; In-t komp'yut. sistem, Odesa, Ekologiya (2018)
24. Tan, F.B., Hunter, G.M.: The repertory grid technique: a method for the study of cognition in information systems. *MIS Q.* **26**(1), 39–57 (2002). <https://doi.org/10.2307/4132340>



IaaS-Application Development for Paralleled Remote Sensing Data Stream Processing

Vadym Zhernovyi¹ , Volodymyr Hnatushenko²  , and Olga Shevtsova² 

¹ Oles Honchar Dnipro National University, Dnipro, Ukraine
vadim.zhernovoy@gmail.com

² Dnipro University of Technology, Dnipro, Ukraine
vvgnat@ukr.net, shevtsova.o.s@nmu.one

Abstract. During the last decade, huge advancements took place in remote sensing affecting speed and quality of data acquisition, processing and mapping. This rapid progress became possible especially due to shifts in software development involving cloud technologies as well as success of deep neural networks (DNN) in processing tremendous amount of data. In particular tasks such as image processing, DNNs achieve remarkable results considering accuracy and precision of pattern recognition. In order to achieve human-level performance solving image processing problem, Deep Learning (DL) solutions must be appropriately trained using powerful hardware worth of thousands of dollars which is often suboptimal to handle using on-premises servers. Current work suggests infrastructure-as-a-service (IaaS) solution for landing remote sensing (RS) data stream processing algorithm involving the latest advancements in image processing using DL solutions and cloud technologies optimized for work with the suggested algorithm such as Kubernetes and Apache Airflow hosted on Google Computing Platform (GCP). The suggested algorithm is represented as a directed acyclic graph (DAG) within IaaS-application. The mentioned cloud technologies are used for better representation of the workflow which implements a complex system for paralleling compute-heavy tasks of very high resolution (VHR) satellite imagery processing to provide the visualized segmentation results for urban development maps in a fast and efficient way.

Keywords: Cloud Computing · Infrastructure-as-a-service · Remote sensing · Deep learning · Urban development · Directed acyclic graph · Google Computing Platform · Kubernetes · Airflow

1 Introduction

RS Imagery still remains a field of various unresolved problems considering efficiency of pattern recognition. By efficiency, the broad term is meant involving

data quality, speed of processing, accuracy, etc. Last decade, neural network are challenged to resolve the most of them propagating DL approach. DNNs keep popularity as a very effective tool set to solve image processing problems in particular. Few of the major obstacles for DL remain labeled data scarcity as well as cost and speed of computation.

If data scarcity can be partially resolved by various approaches such as augmentation, computation tasks still require tremendous efforts to set it up effectively for DL tasks. Modern DNN solutions which are considered successful in solving image processing and pattern recognition problems are usually nearly impossible to train without help of expensive graphics processing units (GPU). Cloud Computing (CC) made it available to use power of multiple dozens computers on demand to accomplish neural network training and image processing task in a fast and efficient way. CC implements a variety of tools to seamlessly schedule and run computation-heavy tasks. GCP in particular allow developers to implement a framework for automatically create an infrastructure in cloud which meets the requirements for effective neural network functioning and scale it correspondingly for either training or inference needs.

Current work suggest a solution in a form of IaaS-application to solve multiple problems mentioned above. With aid of Kubernetes and Apache Airflow, the algorithm has been built to effectively train and run a DNN for the needs of continuous RS data stream processing.

2 Problem Statement

Data is the most crucial part of any Machine Learning (ML) problem. Along with common data problems such as data quality and quantity, labeled data scarcity, etc. [14, 23], there are additional problems that add up specific to RS imagery, namely intraclass diversity, large variance and so on [1, 16]. Unavailability of Big Data in the vast majority of cases leads to overfitting in the most popular and reliable DNN solutions where they have near 100 percent accuracy in training dataset and demonstrate horrible nearly random accuracy of pattern recognition in the real world data. An additional limitation related to RS imagery is that even if the network is trained properly (not overfitted), it still will most likely fail on imagery data registered by other satellite vehicles it was not trained on. When solving RS image processing problems, two parts of data quality concerns should be considered: general ones and one related specifically to the imagery data for a particular satellite vehicle of interest [3].

Except data, another piece of problem which complicates training and inference of good RS image recognition model is an infrastructure [21]. In order to train a good neural network, it becomes nearly impossible to do so using CPUs so the only option to perform training and tuning of neural network models in a reasonable amount of time is exploitation of GPUs [17, 22]. The overwhelming majority of operations performed on imagery data is 2D convolution which is

defined as follows:

$$y[m, n] = x[m, n] * h[m, n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i, j] \cdot h[m - i, n - j] \quad (1)$$

where x represents an image matrix of size $m \times n$, h is a convolution kernel matrix to be applied to the original image to form an output result y . i and j are indices of pixels in the original image x .

Due to the effectiveness of modern GPUs performing multiplication and parallel computing [4, 18], their performance exponentially grows in comparison to CPU considering a complexity of an operation $O(MNkk)$, where a size of an image is $M \times N$ and a size of a convolution kernel is $k \times k$. Table 1 demonstrates the difference in speed between CPU and GPU performing a convolution operation 10 times using $7 * 7$ kernel on 100 RGB images of different size.

Table 1. Comparison between CPU and GPU performing 10 convolution operations on RGB images

Image num.	Image size	GPU (sec)	CPU (sec)	GPU/CPU
100	$128 \times 128 \times 3$	0.04023069699996995	7.1446102779998455	177x
200	$256 \times 256 \times 3$	0.47230450900019605	51.39644785900009	108x
500	$256 \times 256 \times 3$	0.8941660059999776	125.00879671299981	139x

However, considering huge cost of GPUs, especially due to the peak of popularity of cryptocurrency mining [19], it is a problem of utmost importance to make use of GPUs as efficient as possible.

3 Literature Review

When looking for the optimal neural network solution, it is essential to consider multiple options for the neural network structure as well as target models for tuning and further optimizations. One of the most important problem to solve is to be able to train and tune a successful neural network model with limited data, limited resources and limited model complexity [12] to minimize overall cost of the process [25, 26].

One of the promising approaches for DNN training and tuning is distributed deep neural networks (DDNN) which are designed to operate within docker orchestration frameworks such as Kubernetes or Docker Swarm [24, 27, 31]. These orchestration frameworks allows users to create workloads to meet particular needs. Using DDNN approach it is possible to setup DNN training workflow in a cost-efficient way so the most of hardware resources available are used in the most predictable way. Paper [33] suggests maintaining a docker container with lightweight analytical neural network for better management of DDNN training process. Extensive prototype experiments on Amazon web services shew that

such methods can demonstrate predictable training performance while reducing the monetary cost for DDNN workloads with acceptable runtime overhead [33].

Another way to make DNN training process more effective is dynamic precision schedules which are designed to optimize training process for hitting a particular goals such as speed-accuracy trade-offs. One of the direction of such optimizations relevant to current work is lowering computation cost [32]. Low-precision DNN training has gained tremendous attention as reducing precision is one of the most effective knobs for boosting DNNs' training time/energy efficiency [6].

Quantized methods of backpropagation are tended to be promising for improving overall DNN training process which may lead to more efficient DNN training process. One of the main sources of time and energy drains is the well known backpropagation (backprop) algorithm, which roughly accounts for 2/3 of the computational cost of training [29].

There are several DNN-agnostic techniques to leverage temporal and spatial multiplexing to improve GPU utilization for deep learning inference workloads [15].

Approaches described above demonstrate that optimization is possible on nearly any level of DNN training:

- infrastructure level (docker, cluster)
- neural network level (dynamic precision schedules, quantized backprop)
- hardware level (GPU, CUDA, TensorRT)

The objective of the research is to develop an algorithm to create and schedule a workflow in GCP IaaS-platform for cost-efficient use of cloud resources to support an infrastructure for Remote Sensing image recognition neural network training and inference.

4 Algorithm Development

Algorithm requires to cover the most aspects of model training and tuning leveraging all computation resources available and to be able to scale training/tuning environment appropriately:

- it must be ready to start training whenever new data is available;
- it must allocate sufficient resources for the task;
- it must schedule corresponding amount of work and make sure all allocated resources are used most of the time;

In order to implement these requirements, in current work Apache Airflow [28] and Google Kubernetes Engine [2] are used. Airflow is a community-driven set of tools which makes it possible to implement various workflow and any kind of scheduling for it. Airflow implements a message queue in order to orchestrate certain tasks. A workflow within this framework is implemented in a form of DAG. In addition to this, Airflow allows to parameterize DAGs and can be extended with plugins to introduce support for any kind of other services. In

current work Airflow is used in conjunction with Kubernetes implemented within GCP.

Kubernetes itself is a system for automated management and scaling of containerized applications. On practice, it allows users to organize certain hardware resources into a cluster. Cluster itself doesn't operate any resources and only manages metadata about resources available. Cluster always consists of one or more node pools which are representation of certain amount of hardware resources. These node pools can be scaled according to the rules set within this particular node pool. For current research this property of node pools is leveraged respecting a particular needs for DDNN training and tuning.

4.1 Data Loading and Batch Calculation

For the purpose of this algorithm, data is stored within a storage and can be accessed for RS image processing when necessary. This storage is managed by an operator which marks labeled data with corresponding status:

- *pending* - default state for any new data;
- *processing* - data is scheduled for work;
- *done* - data is used for training purposes;

The storage is scanned on certain timeout. The whole data management process can be defined as predicate expression:

$$\forall x \forall t \exists s \in S, P(x, t, s) \quad (2)$$

where x is an image data object, t is a period of time and s is one of the statuses from a discrete domain of statuses $S = \{pending, processing, done\}$.

4.2 Kubernetes Cluster and Workload Calculation

For the most cases, Kubernetes cluster (KC) is created and maintained to provide stable and scalable platform for the needs of different web services. However, for DL tasks, such workflow is not optimal due to the nature of DL training and tuning which require a rapid spike in resource use once in awhile. The suggested algorithm implements a workflow which doesn't require KC to be up all the time. Our approach is based around data storage. The operator defined keeps track of the storage and creates a cluster on these demands.

KC is designed to be created with a number of training and tuning (TT) node pools equal to the number of neural network architectures used in the experiment. This decision is based on different hardware resource needs of every particular network mostly due to the size of the network and not architecture. KC itself doesn't bring anything specific for the algorithm, the node pools configuration is the only part that matters for work of algorithm.

The node pools are configured in a certain way to support a concept of distributed training with DDNN as well as protected by a virtual private network (VPN) to prevent unnecessary exposure to the internet. Kubernetes provide node

pool options for automatic scaling which specifically configured for the algorithm needs. The way automatic scaling works is the specific daemon called autoscaler provisions additional nodes where each node is a representation of one computer with certain amount of CPU, GPU, memory and disk resources. The decision on scaling are made by the following equation:

$$R_{i+1} = \lceil R_i * (C_m) / (D_m) \rceil, i \in N, m = \{cpu, gpu, mem\} \quad (3)$$

where R_i and R_{i+1} are a current and desired replica of certain resources within i -th node pool, (C_m) and (D_m) are a current and desired metric value which is defined by a tuple m . It is worth mentioning that within each node pool only nodes of one certain type can be created - this is one of the main resources why separate node pools are created for different types of DNN. However, for the purpose of current work, the auto scaling model is adjusted to make sure that all available resources are used most of the time, especially GPU:

- all requested resources are allocated immediately;
- minimum number of nodes is 0 instead of 1;
- vertical scaling is introduced.

When training a model, it is expected that all requested resources will be available right away considering a particular node specification within the node pool:

$$N_i = \lceil D_m / S_{m,i} \rceil, i \in N, m = \{cpu, gpu, mem\} \quad (4)$$

where N_i is an amount of nodes to allocate in i -th node pool, D_m is an amount of m -th resource requested and $S_{m,i}$ is a node hardware specification of m -th resource within i -th node pool. When the amount of resources required is 0, all nodes will be deallocated introducing near 0 cost of using a particular node pool when there are no training or tuning tasks are available (all task statuses in the data storage are *done*). Vertical scaling is introduced in order to avoid idling of resources. When a particular subnet of DDNN is finished its tasks, the freed resources are split among remaining running processes.

4.3 Workflow Scheduling

In order to make data stream and cluster auto scaling work, a workflow must be defined. The best way to describe the algorithm is a dynamic directed acyclic graph (DAG). In order to update data status cyclic dependency is not needed since the original data is read only and not changed in any way and data statuses are managed separately considering any missing values in the result as *pending*. Data is revised each time the training process is started. There are multiple reasons DAG is dynamic: some of its vertices represent a node pool creation process and an amount of DNN models and their specifications may vary. Another vertices represent distributed training process when each DNN are split into certain amount of subnetworks (Fig. 1). So the final DAG is calculated considering an amount if neural network specifications and an amount of subnetworks for each network. For each subnetwork the pod is created representing a particular subnetwork training process. Pod, in particular, defines an amount of resources needed for any task.

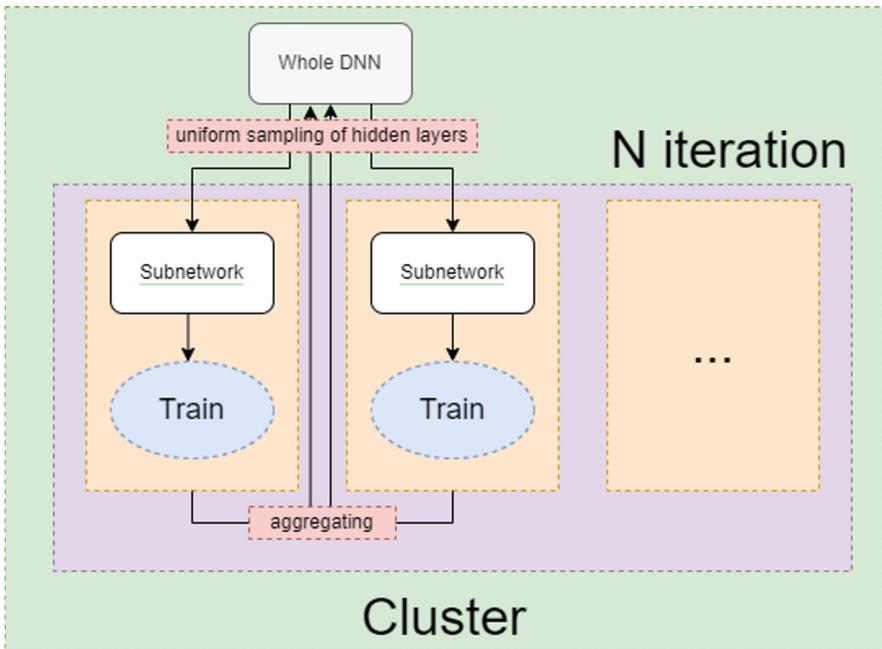


Fig. 1. Split of DNN model among available resources within a computation cluster

For demonstration of the suggested approach flexibility, an additional vertices are introduced which are supplemental to DL training process. When all pods are terminated (which means all tasks are done), the 5-fold cross-validation (CV) is introduced with a subsequent evaluation of the best trained model of each type.

The final vertex of the DAG is a cluster disposal. When all processes are done, training results are saved to the persistent storage outside of the cluster, so the cluster is not needed anymore.

The whole dynamic DAG scheme is on a Fig. 2.

5 Experiment, Results and Discussion

For this experiment multiple tools are used, the main of them is Apache Airflow. It is open source community-driven framework which is designed for scheduling and monitoring of workflows and DAG is considered as a foundation of this tool to define any workflows which naturally fits with the suggested algorithm. Each element of the DAG in Airflow is defined by operators. These operator are implemented to represent a part of functionality of many different platform including Kubernetes. There are Kubernetes operator for managing cluster and pods which are sufficient for algorithm implementation.

Google Kubernetes Engine (GKE) is used due to availability vertical scaling approach and fees coverage whenever cluster resources are not used which makes

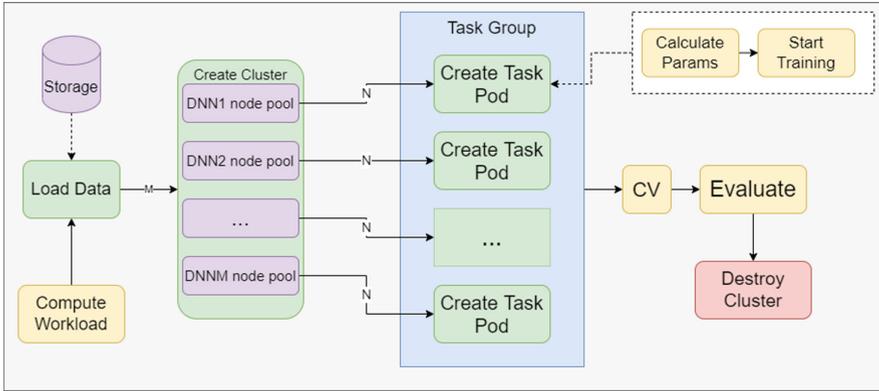


Fig. 2. Training algorithm DAG

cost being dependent only from particular hardware resources used since the algorithm is designed to use 99% percent of these resource 100% of the time.

To avoid charges beyond a trial policy within Google Cloud Platform, the cheapest GPU configurations are used for node specification which is n1-standard-1 machine type with NVIDIA Tesla T4 GPU.

5.1 Neural Networks

When designing a neural network to use on RS imagery data [10], there multiple concerns arise:

- big intraclass diversity;
- high interclass similarity (also known as low between-class separability);
- large variance of object/scene scales;

Nowadays, there are many options to pick from considering the best model for a particular data [10, 13]. Since the private image dataset made of WorldView-3 data is used in the current work, it is not widely tested with different approaches. Using the suggested algorithm, it is fluent to conduct an experiment to figure out the best model among many candidates. When all candidates are chosen, the DAG is dynamically reconstructed to represent this

The latest meta-analysis on DNN application for Remote Sensing suggest the best models based on a public datasets performance [7]. For the purpose of this research, a list of 4 CNN models is conducted heuristically to demonstrate work of the algorithm:

- attention recurrent convolutional network (ARCNet);
- multiscale CNN (MCNN);
- scale-free CNN (SF-CNN);
- gated bidirectional network (GBN);

Due to relatively scarce data, all models leverage transfer learning in order to implement CNN base which serves as a feature extractor developed within [11].

For each of 4 neural network models, in addition to categorical cross entropy loss function, an additional cost effective hierarchical Wasserstein loss [5] function for discriminative CNN is used [30]. The D-CNNs shows better discriminability in scene classification.

5.2 Experimental Datasets

Due to scarcity of labeled data available in current work, the dataset is used which was developed in the previous work [9]. The dataset is based on WorldView-3 satellite VHR imagery. For purposes of achieving better DNN model performance, additional augmentation were implemented leveraging unique properties of satellite imagery like near infra-red (NIR) channels. Pansharpening technique were applied to fuse VHR monochrome image with multispectral one of lower resolution [8]. Active learning techniques are used to make dataset more diverse overtime [20] The dataset used in current work is an object-level segmentation dataset (Fig. 3).



Fig. 3. An example of object-level segmentation labels

5.3 Simulation Results, Computation Cost and Discussion

Metric that is used for measuring performance of the models is overall accuracy (OA):

$$OA = N_c/N_t \quad (5)$$

where N_c is a number of accurately classified samples and N_t is a number of test samples.

Train/Test ratio used is 90/10 where 90% of data are used in 5-fold CV.

For calculating computation cost, time is considered an adequate metric since the algorithm is designed to be hardware-agnostic and can be adjusted regarding hardware to meet particular needs so the cost of cloud hardware resource is represented by X . Furthermore, the auto scaling schema implemented justifies the statement that an amount of resource with a cost of X per hour are used by 100% all the time. To justify the use of the suggested algorithm costs are compared to the standard approach:

- standard cluster settings;
- default DNN loss functions;
- no distributed training (1 GPU).

The cost efficiency is calculated as a ratio of computation costs:

$$CE = \frac{C_{alg}}{C_{std} * N_{gpu}} * 100\% \quad (6)$$

where C_{alg} and C_{std} are costs calculated using the suggested algorithm and the standard approach mentioned above and N_{gpu} is a number of GPU used in the algorithm due to distributed training.

The Table 2 demonstrates the results of simulation using the suggested algorithm.

Table 2. Model training efficiency results

Model name	Loss function	OA	GPU used	Computation cost (X-hours)	CE (%)
ARCNet	categorical	87.5%	4	7.1446102779998455	122
ARCNet	wasseratein	91.02%	4	51.39644785900009	121
MCNN	categorical	79.54%	2	125.00879671299981	126
MCNN	wasseratein	81.44%	2	7.1446102779998455	125
SF-CNN	categorical	85.5%	3	51.39644785900009	131
SF-CNN	wasseratein	89.9%	3	125.00879671299981	141
GBN	categorical	83.7%	2	7.1446102779998455	111
GBN	wasseratein	87.2%	2	51.39644785900009	112

6 Conclusions

In current article, the simple algorithm is developed which main purpose is to mitigate the sophisticated process of large scale neural network training. The suggested end-to-end algorithm is flexible in many ways so it can be configured to use any number of ML models with data managed implicitly.

Despite many concern remain to be improved in RS imagery processing, namely data issues and better ML solutions, there are also consequential problems of large scale training with existing data and models which this work is willing to solve - computation costs. Deep neural networks become more complex and more precise but still at a huge cost of training and tuning. The problem escalates quickly when it comes to train multiple models in order to figure out better one and here cost of computation raises very quickly.

Many solutions considering decrease costs complexities often target more granular parts of the problem like better scaling formula or another DNN architecture. The suggested algorithm is rather a complex end-to-end solution which is ready to use with all tools being available and open source. When the algorithm is being complex in details, the described simulation approach demonstrates that it is very easy to setup and run due to straight-forward definition of the algorithm in a form of DAG in Apache Airflow infrastructure with all Kubernetes adjustments managed transparently.

The simulation experiment shows that it is possible to setup any configuration regarding models and resources available and there is a clear evidence of better cost management according to CE metric developed which reflects the algorithm efficiency adequately. More models and cluster configuration are yet to be tested to figure out the least possible cost efficiency improvement using the suggested solution. However, according to experiment during this research, it seems to be possible to achieve at least 10% costs improvement using this algorithm, when for the most models CE reaches beyond 20% improvement.

Future work in this direction is to conduct more experiments to inspect the impact of every part of the solution and their correlation to identify the room for further optimization and to aggregate more valuable improvements for improving training and tuning cost efficiency.

References

1. Baret, F., Buis, S.: Estimating canopy characteristics from remote sensing observations: review of methods and associated problems. *Adv. Land Remote Sens.* pp. 173–201 (2008). https://doi.org/10.1007/978-1-4020-6450-0_7
2. Bisong, E.: Containers and google kubernetes engine. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 655–670. Apress, Berkeley, CA (2019). https://doi.org/10.1007/978-1-4842-4470-8_45
3. Bruzzone, L., Demir, B.: A review of modern approaches to classification of remote sensing data. In: Manakos, I., Braun, M. (eds.) *Land Use and Land Cover Mapping in Europe*. RSDIP, vol. 18, pp. 127–143. Springer, Dordrecht (2014). https://doi.org/10.1007/978-94-007-7969-3_9

4. BUBER, E., DIRI, B.: Performance analysis and CPU vs GPU comparison for deep learning. In: 2018 6th International Conference on Control Engineering Information Technology (CEIT), pp. 1–6 (2018). <https://doi.org/10.1109/CEIT.2018.8751930>
5. Frogner, C., Zhang, C., Mobahi, H., et al.: Learning with a Wasserstein loss. In: Advances in Neural Information Processing Systems, vol. 28 (2015). <https://doi.org/10.48550/arXiv.1506.05439>
6. Fu, Y., Guo, H., Li, M., et al.: Cpt: efficient deep neural network training via cyclic precision. arXiv preprint [arXiv:2101.09868](https://arxiv.org/abs/2101.09868) (2021). <https://doi.org/10.48550/arXiv.2101.09868>
7. Ghanbari, H., Mahdianpari, M., Homayouni, S., Mohammadimanesh, F.: A meta-analysis of convolutional neural networks for remote sensing applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **14**, 3602–3613 (2021). <https://doi.org/10.1109/JSTARS.2021.3065569>
8. Hnatushenko, V., Hnatushenko, V., Kavats, O., et al.: Pansharpening technology of high resolution multispectral and panchromatic satellite images. *Sci. Bull. Nat. Min. Univ.* **4**, 91–98 (2015)
9. Hnatushenko, V., Zhernovyi, V.: Complex Approach of High-Resolution Multi-spectral Data Engineering for Deep Neural Network Processing. In: Lytvynenko, V., Babichev, S., Wójcik, W., Vynokurova, O., Vyshemyrskaya, S., Radetskaya, S. (eds.) ISDMCI 2019. AISC, vol. 1020, pp. 659–672. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-26474-1_46
10. Hnatushenko, V., Zhernovyi, V.: Method of improving instance segmentation for very high resolution remote sensing imagery using deep learning. In: Babichev, S., Peleshko, D., Vynokurova, O. (eds.) DSMP 2020. CCIS, vol. 1158, pp. 323–333. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61656-4_21
11. Hnatushenko, V., Zhernovyi, V., Udovyk, I., Shevtsova, O.: Intelligent system for building separation on a semantically segmented map. In: CEUR Workshop Proceedings, pp. 1–11 (2021)
12. Hong, S., Roh, B., Kim, K.H., et al.: PvaNet: lightweight deep neural networks for real-time object detection. arXiv preprint [arXiv:1611.08588](https://arxiv.org/abs/1611.08588) (2016). <https://doi.org/10.48550/arXiv.1611.08588>
13. Hordiiuk, D., Hnatushenko, V.: Neural network and local laplace filter methods applied to very high resolution remote sensing imagery in urban damage detection. In: 2017 IEEE International Young Scientists Forum on Applied Physics and Engineering (YSF), pp. 363–366 (2017). <https://doi.org/10.1109/YSF.2017.8126648>
14. Hutchinson, M., Antono, E., Gibbons, B., et al.: Overcoming data scarcity with transfer learning. arXiv preprint [arXiv:1711.05099](https://arxiv.org/abs/1711.05099) (2017). <https://doi.org/10.48550/arXiv.1711.05099>
15. Jain, P., Mo, X., Jain, A., Subbaraj, H., et al.: Dynamic space-time scheduling for GPU inference. arXiv preprint [arXiv:1901.00041](https://arxiv.org/abs/1901.00041) (2018). <https://doi.org/10.48550/arXiv.1901.00041>
16. Kaab, A.: Remote sensing of permafrost-related problems and hazards. *Permafrost Periglac. Process.* **19**(2), 107–136 (2008). <https://doi.org/10.1002/ppp.619>
17. Kimmel, J., Mcdole, A., Abdelsalam, M., Gupta, M., Sandhu, R.: Recurrent neural networks based online behavioural malware detection techniques for cloud infrastructure. *IEEE Access* **9**, 68066–68080 (2021). <https://doi.org/10.1109/ACCESS.2021.3077498>

18. Madiajagan, M., Raj, S.: Parallel computing, graphics processing unit (GPU) and new hardware for deep learning in computational intelligence research. In: Deep Learning and Parallel Computing Environment for Bioengineering Systems, pp. 1–15. Elsevier (2019). <https://doi.org/10.1016/B978-0-12-816718-2.00008-7>
19. Mueller, P.: Cryptocurrency mining: asymmetric response to price movement. Available at SSRN 3733026 (2020). <https://doi.org/10.2139/ssrn.3733026>
20. Natarajan, A., Ganesan, D., Marlin, B.: Hierarchical active learning for model personalization in the presence of label scarcity. In: 2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 1–4. IEEE (2019). <https://doi.org/10.1109/BSN.2019.8771081>
21. Ranjit, M.P., Ganapathy, G., Sridhar, K., Arumugham, V.: Efficient deep learning hyperparameter tuning using cloud infrastructure: Intelligent distributed hyperparameter tuning with Bayesian optimization in the cloud. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), pp. 520–522 (2019). <https://doi.org/10.1109/CLOUD.2019.00097>
22. Sethi, K., Kumar, R., Prajapati, N., Bera, P.: Deep reinforcement learning based intrusion detection system for cloud infrastructure. In: 2020 International Conference on Communication Systems Networks (COMSNETS), pp. 1–6 (2020). <https://doi.org/10.1109/COMSNETS48256.2020.9027452>
23. Sovrano, F., Palmirani, M., Vitali, F.: Combining shallow and deep learning approaches against data scarcity in legal domains. *Gov. Inf. Quart.* **39**(3), 101715 (2022). <https://doi.org/10.1016/j.giq.2022.101715>
24. Strom, N.: Scalable distributed DNN training using commodity GPU cloud computing. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
25. Sze, V., Chen, Y.H., Yang, T.J., Emer, J.: Efficient processing of deep neural networks: a tutorial and survey. *Proc. IEEE* **105**(12), 2295–2329 (2017). <https://doi.org/10.1109/JPROC.2017.2761740>
26. Sze, V., Chen, Y.H., Yang, T.J., Emer, J.: Efficient processing of deep neural networks. *Synth. Lect. Comput. Archit.* **15**(2), 1–341 (2020). <https://doi.org/10.2200/S01004ED1V01Y202004CAC050>
27. Teerapittayanon, S., McDanel, B., Kung, H.T.: Distributed deep neural networks over the cloud, the edge and end devices. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 328–339. IEEE (2017). <https://doi.org/10.1109/ICDCS.2017.226>
28. Walter-Tscharf, V.: Implementation and evaluation of a MLaaS for document classification with continuous deep learning models. In: Architecture, Engineering, and Technology (AET), p.55
29. Wiedemann, S., Mehari, T., Kepp, K., Samek, W.: Dithered backprop: A sparse and quantized backpropagation algorithm for more efficient deep neural network training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 720–721 (2020). <https://doi.org/10.48550/arXiv.2004.04729>
30. Yao, Y., Deng, J., Chen, X., et al.: Deep discriminative CNN with temporal ensembling for ambiguously-labeled image classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12669–12676 (2020). <https://doi.org/10.1609/aaai.v34i07.6959>
31. You, Y., Zhang, Z., Hsieh, C.J., Demmel, J., Keutzer, K.: Fast deep neural network training on distributed systems and cloud tpus. *IEEE Trans. Parallel Distrib. Syst.* **30**(11), 2449–2462 (2019). <https://doi.org/10.1109/TPDS.2019.2913833>

32. Yu, Z., Fu, Y., Wu, S., et al.: Ldp: learnable dynamic precision for efficient deep neural network training and inference. arXiv preprint [arXiv:2203.07713](https://arxiv.org/abs/2203.07713) (2022). <https://doi.org/10.48550/arXiv.2203.07713>
33. Zheng, H., Xu, F., Chen, L., Zhou, Z., Liu, F.: Cynthia: Cost-efficient cloud resource provisioning for predictable distributed deep neural network training. In: Proceedings of the 48th International Conference on Parallel Processing, pp. 1–11 (2019). <https://doi.org/10.1145/3337821.3337873>



Correction to: Classification Methods of Heterogeneous Data in Intellectual Systems of Medical and Social Monitoring

Olena Arsirii , Svitlana Antoshchuk , Olga Manikaeva ,
Oksana Babilunha , and Anatolii Nikolenko 

Correction to:

**Chapter “Classification Methods of Heterogeneous Data
in Intellectual Systems of Medical and Social Monitoring” in:
S. Babichev and V. Lytvynenko (Eds.): *Lecture Notes in Data
Engineering, Computational Intelligence, and Decision Making*,
LNDECT 149, https://doi.org/10.1007/978-3-031-16203-9_38**

In the original version of the book following belated correction has been incorporated: Reference 6 has been revised in Chapter “Classification Methods of Heterogeneous Data in Intellectual Systems of Medical and Social Monitoring”. The book and the chapter have been updated with the changes.

The updated original version of this chapter can be found at
https://doi.org/10.1007/978-3-031-16203-9_38

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
S. Babichev and V. Lytvynenko (Eds.): ISDMCI 2022, LNDECT 149, p. C1, 2023.
https://doi.org/10.1007/978-3-031-16203-9_40

Author Index

A

Antoshchuk, Svitlana, 511, 686
Arsirii, Olena, 686
Azarova, Larisa, 333

B

Babichev, Sergii, 3, 25
Babilunha, Oksana, 686
Baklan, Igor, 426
Balanda, Anatolii, 61
Bardachov, Yuriy, 387
Barmak, Oleksander, 591
Barmak, Olexander, 333
Batryn, Natalia, 488
Berezsky, Oleh, 488
Bidyuk, Peter, 214
Boby, Yuliia, 61
Borzov, Yuriy, 639
Boskin, Oleg, 387
Boskin, Oleh, 160
Boyko, Nataliya, 126
Bunyak, Yuriy, 177
Burak, Nazarii, 639
Burov, Yevhen, 608
Bychkov, Alexey, 266

C

Chand, Roneel, 231
Chyrun, Lyubomyr, 286, 608

D

Dereko, Vladyslav, 449
Derysh, Bohdan, 488
Durnyak, Bohdan, 3, 25

F

Filatova, Anna, 320

G

Gado, Iryna, 25
Golovatyi, Roman, 367
Gonchar, Olga, 197
Goncharenko, Tatiana, 3
Gozhyj, Aleksandr, 105, 214

H

Hnatushenko, Viktoriia, 525
Hnatushenko, Volodymyr, 407, 664, 705
Hodovychenko, Mykola, 511
Hong, Tzung-Pei, 564
Hu, Zhengbing, 564

I

Ilyin, Oleh, 160

K

Kalahnik, Ganna, 266
Kalinina, Irina, 105, 214
Karabyn, Oksana, 148
Karabyn, Vasy, 148
Kashtan, Vita, 664

Kashyap, Esha, 564
 Khlevnoi, Oleksandr, 639
 Kirichenko, Lyudmyla, 547
 Kochura, Tetiana, 61
 Kondratiuk, Serhii, 333
 Kordunova, Yuliia, 367
 Korobchynskyy, Maksym, 449
 Kovalchuk, Oleksii, 591
 Kovtun, Oleksandr, 449
 Kozhushko, Olena, 74
 Krak, Iurii, 333, 591
 Krapivina, Galina, 197
 Kravets, Petro, 286
 Kudriashova, Alona, 246
 Kuznetsov, Vladyslav, 333
 Kuzyk, Andrij, 148
 Kyvetyny, Roman, 177

L

Last, Mark, 564
 Liakh, Igor, 25
 Liashchynskyy, Petro, 488
 Liubchuk, Olha, 197
 Lobachev, Ivan, 511
 Lobachev, Mykhaylo, 511
 Louda, Petr, 462
 Lysa, Natalya, 353
 Lytvyn, Vasyl, 286, 608

M

Malets, Igor, 148
 Manikaeva, Olga, 686
 Martsyshyn, Roman, 353
 Martyniuk, Petro, 74
 Mashkov, Oleh, 160, 266
 Mashkov, Viktor, 160
 Matsuki, Yoshio, 214
 Matviichuk, Myroslav, 92
 Mazurets, Olexander, 591
 Miyushkovych, Yuliya, 353
 Molchanova, Maryna, 591
 Moshynskyy, Viktor, 74
 Mushtat, Oleksandr, 651

N

Nadruga, Vasiliy, 61
 Nikolenko, Anatolii, 686
 Novikov, Vsevolod, 387
 Novikova, Anastasiia, 387

O

Obydenyi, Yevhen, 651
 Ohnieva, Oksana, 160

P

Padiuk, Pavlo, 333
 Pavlenko, Kyrylo, 547
 Petrushenko, Natalia, 197
 Pichugina, Oksana, 547
 Pikh, Iryna, 246
 Pitsun, Oleh, 488
 Polodiuk, Mariana, 61
 Polyvoda, Oksana, 42
 Polyvoda, Vladyslav, 42
 Povoroznyuk, Anatolii, 320
 Povoroznyuk, Oksana, 320
 Prydatko, Oleksandr, 367

R

Radivilova, Tamara, 547
 Raghuvaiya, Krishna, 231
 Raita, Diana, 639
 Romanyuk, Olexander, 302
 Roskladka, Andrii, 302
 Roskladka, Nataliia, 302
 Rozov, Yuriy, 387
 Rudenko, Myhailo, 449

S

Safonyk, Andrii, 92
 Sarnatskyi, Vladyslav, 426
 Savina, Nataliia, 591
 Savytska, Liudmyla, 302
 Schöler, Thorsten, 511
 Senkivska, Nataliia, 246
 Senkivskyy, Vsevolod, 246
 Shakhina, Iryna, 320
 Sharko, Artem, 462
 Sharko, Marharyta, 197
 Sharko, Oleksandr, 462
 Shedlovska, Yana, 407
 Shedlovsky, Igor, 407
 Shevchenko, Victor, 266
 Shevtsova, Olga, 705
 Shostak, Liubov, 74
 Sikora, Lubomyr, 353
 Slobodzian, Vitalii, 591
 Smotr, Olga, 148, 367
 Sobko, Olena, 591
 Sofina, Olga, 177
 Sokolova, Natalya, 651
 Soldatenko, Dmytro, 525
 Stepanchenko, Olha, 74
 Stepanchikov, Dmitry, 462

T

Troianovska-Korobeinikova, Tetiana, 302
 Tyshchenko, Oleksii K., 564
 Tyshchenko, Oleksii, 387

V

Vanualailai, Jito, [231](#)
Vasylenko, Nataliia, [197](#)
Vorobyova, Kateryna, [197](#)
Voronenko, Mariia, [387](#)
Vyshemyrska, Svitlana, [266](#), [387](#)
Vysotska, Victoria, [286](#), [608](#)

Y

Yasinska-Damri, Lyudmyla, [3](#)

Z

Zaitsev, Oleksandr, [449](#)
Zhernovyi, Vadym, [705](#)
Zhuravlova, Yuliia, [651](#)