Studies in Systems, Decision and Control 547

Gebrail Bekdaş
Sinan Melih Nigdeli *Editors*

New Advances in Soft Computing in Civil Engineering

AI-Based Optimization and Prediction

Springer

Studies in Systems, Decision and Control

Volume 547

The series "Studies in Systems, Decision and Control" (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control–quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution and exposure which enable both a wide and rapid dissemination of research output.

Indexed by SCOPUS, DBLP, WTI Frankfurt eG, zbMATH, SCImago.

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

Gebrail Bekdaş · Sinan Melih Nigdeli
Editors

New Advances in Soft Computing in Civil Engineering

AI-Based Optimization and Prediction

Springer

Editors
Gebrail Bekdaş
Muhendislik Fakultesi
Istanbul Universitesi-Cerrahpasa
Avcilar, Istanbul, Türkiye

Sinan Melih Nigdeli
Muhendislik Fakultesi
Istanbul Universitesi-Cerrahpasa
Avcilar, Istanbul, Türkiye

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-031-65975-1 ISBN 978-3-031-65976-8 (eBook)
https://doi.org/10.1007/978-3-031-65976-8

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

Preface

In this book, several reviews and applications are given for soft computing applications in civil engineering. The use of artificial intelligence (AI) algorithms and machine learning (ML) shows a great increase in the area of civil engineering.

Firstly, a chapter about data analyses and machine learning is given including a review. Then, applications of AI are reviewed in Chapter "Application of Artificial Intelligence (AI) in Civil Engineering" of the book. Chapter "Machine Learning Applications in Structural Engineering" includes reviews of the application of ML in structural engineering and an example is demonstrated in the chapter. Chapter "A Multi-objective Optimal Design Process for Determination of Link Capacity Expansions" is an optimization study for the determination of link capacity expansions. Chapter "IoT with Deep Learning in Pipeline and Metro Track Risk Estimation Using Smart Cities Development" is a deep learning application for risk estimation. Forecasting a lake level is presented in Chapter "Forecasting of Lake Level by Soft Computing Approaches" via soft computing techniques. In Chapters "Artificial Intelligence-Driven Structural Health Monitoring: Challenges, Progress, and Applications" and "Optimizing Tuned Mass Damper by Examining Displacement Ratios with and Without TMD System", a structural health monitoring and a structural control study are given, respectively. In Chapter "Evaluation of Predictive Models for Mechanical Properties of Earth-Based Composites for Sustainable Building Applications", a predictive AI model for the mechanical properties of earth-based composites is presented. Chapters "Shear Wall Cost Optimization by Employing Harmony Search" to "A Modified Jaya Algorithm for Optimum Design of Carbon Fiber-Reinforced Polymers" include metaheuristic-based optimization and machine learning applications for reinforced concrete structures. Chapters "Prediction of Bi-Linear Strength Envelope of Brazilian Soils Using Machine Learning Techniques" and "Assessment of Unconfined Compressive Strength of Stabilized Soil Using Artificial Intelligence Tools: A Scientometrics Review" are about ML applications for the mechanical properties of soil. As a review, Chapter "A Review of Deformations Prediction for Oil and Gas Pipelines Using Machine and Deep Learning" includes a study about deformation prediction for

oil and gas pipelines. Chapters "Determination of the Effect of XGBoost's Parameters on a Structural Problem" to "Investigation of the Effect of Initial Parameters on the Performance of Metaheuristic Algorithms on a Structural Engineering Problem" include structural mechanics applications of AI. Chapters "Geospatial Multi-criteria Decision Framework for Municipal Landfill Site Selection in Surat City, India" and "Comparing Classification Algorithms for Predicting Spatial Land Cover via Landscape Indices in Nashik, India" include case studies from India using AI algorithms.

Avcilar, Istanbul, Türkiye
May 2024

Gebrail Bekdaş
Sinan Melih Nigdeli

Contents

Introduction to Data Analysis and Machine Learning Applications in Civil Engineering

Fatih Tarık Sarı, Sinan Melih Nigdeli, Gebrail Bekdaş, and Umit Işıkdağ

Abstract Nowadays, artificial intelligence is widely applied in various fields. Because of its quickness, ease of use, and high accuracy rate, it is being used more frequently. Machine learning is a sub-branch of artificial intelligence. Data sets obtained from data analysis are used in machine learning. Machine learning raises the accuracy rate and keeps getting better thanks to new data sets. The field of civil engineering has begun to apply artificial intelligence as well. As a result, analysis times have gotten shorter, and the accuracy of the results has increased. This paper summarizes data analysis, machine learning, and its applications in civil engineering.

Keywords Artificial Intelligence · Data Analysis · Machine Learning · Artificial Neural Network · Civil Engineering

1 Introduction

Data analysis is a procedure used to get relevant information or to solve problems. The goal of data analysis is to automate the process to minimize human involvement and achieve faster outcomes by utilizing larger datasets. Data analysis yields results that empower organizations or people to make significant and well-informed decisions. The process involves several stages, including data cleansing and organization, data processing, data visualization, data analysis, data interpretation, and generating

F. T. Sarı · S. M. Nigdeli · G. Bekdaş (✉)
Department of Civil Engineering, Istanbul University - Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

U. Işıkdağ
Department of Architecture, Mimar Sinan Fine-Art University, 34427 Beyoglu, Istanbul, Turkey
e-mail: umit.isikdag@msgsu.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_1

conclusions. Data analysis has a wide range of uses across various sectors. It finds applications in several domains, including education, health, economy and finance, business and marketing, science, and research.

Machine learning is considered a sub-branch of artificial intelligence. It is a computer algorithm that has the ability to learn from data to solve a specific problem. It uses the collected data to construct and refine statistical models for problem-solving purposes. Machine learning algorithms analyze data, identify models, and produce results. Subsequently, it improves these results with more data sets and feedback from experience. Machine learning is applied in various domains, including medical diagnosis, autonomous driving, financial prediction, image and audio processing, e-commerce, and retail.

Machine learning is commonly categorized into the subsequent sub-disciplines. The several types of machine learning include supervised learning, unsupervised learning, reinforcement learning, semi-supervised and reinforced learning, and deep learning. Figure 1 shows a chart of the sub-branches of machine learning.

Data analysis and machine learning are closely interconnected. Data analysis involves the preparation of data sets that are utilized for the training of machine learning algorithms. Data analysis typically uses statistical data, graphics, or tables. Machine learning is used to assess and enhance the outcomes of data analysis. Machine learning is an artificial intelligence technique that utilizes acquired data to provide predictions, classifications, or decisions as its output.

Data analysis and machine learning, with their extensive range of applications, are also employed in the field of civil engineering. It is commonly employed in civil

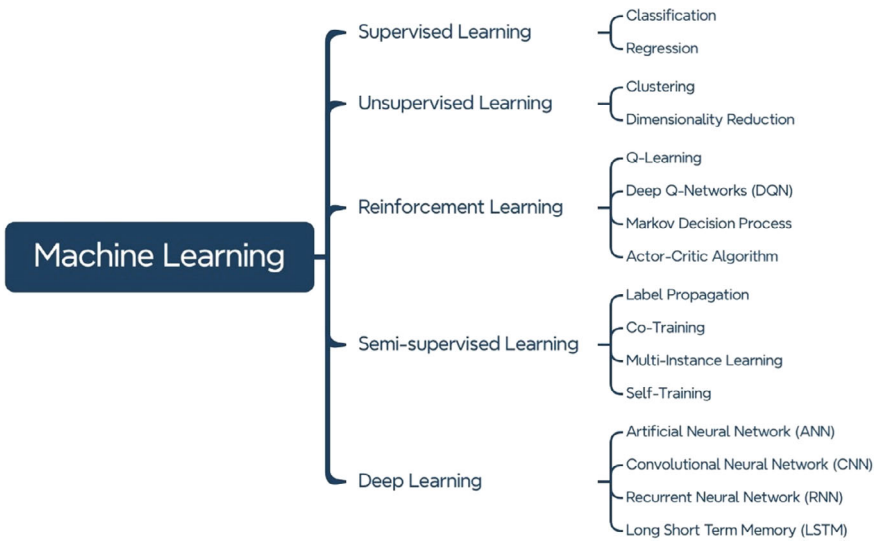


Fig. 1 Sub-branches of machine learning

engineering for tasks such as structural analysis and design, quality control, cost estimation, project management and planning, as well as safety and risk management.

2 Literature Research on Data Analysis and Machine Learning Applications in Civil Engineering

This section includes research that has utilized data analysis and machine learning techniques to investigate various subjects within the domain of civil engineering.

(a) **Material Quality Control**

In the studies carried out for material quality control, there are many studies using machine learning algorithms to predict the compressive strength, modulus of elasticity, and concrete properties and to predict the strength of concretes with different admixtures.

Behnood et al. [18] have studied the prediction of concrete tensile strength using machine learning techniques. They have observed that machine learning techniques accurately predicted concrete tensile strength by outperforming NLR. Behnood et al. [14] have used the M5P model tree algorithm to predict the compressive strength of normal and high-strength concretes. They have collected data from the literature for the created model. They observed that the model accuracy exceeded 80% of the accuracy rate in predicting the compressive strength of concrete. Behnood and Golafshani [17] have predicted the compressive strength of concrete with silica fume using hybrid artificial neural networks. They have used multi-objective grey wolf optimization (GWO). They have concluded that the silica fume-to-binder ratio significantly affected the concrete strength. Young et al. [94] have used data sets obtained from concrete mixtures at the job site and in the laboratory to predict concrete strength and design the optimum mix. Behnood and Golafshani [16] have predicted the properties of concrete containing waste foundry sand using the M5P model tree algorithm. The developed models predict the elastic modulus, compressive, flexural, and tensile strengths of concrete. Golafshani et al. [43] have studied the prediction of the compressive strength of normal and high-performance concrete using ANN, GWO, and ANFIS. They have observed that the ANN model performed better than the ANFIS model. Kandiri et al. [53] have studied the compressive strength prediction of concrete containing ground granulated blast furnace slag using an artificial neural network (ANN) and a salp swarm algorithm (SSA). Gholampour et al. [42] have used artificial intelligence methods for compressive strength prediction of coarse recycled concrete aggregates. They have used MARS, M5Tree, and LSSVR models and observed that the LSSVR model showed improved accuracy in compressive strength prediction. Han et al. [48] have developed a machine learning algorithm for the prediction of the modulus of elasticity of RCA concretes with recycled aggregates and optimum mix designs. Feng et al. [39] have analyzed concrete properties for compressive strength prediction of concrete using machine learning. They have concluded that AdaBoost predicts concrete strength with 98% accuracy. Feng et al.

[40] have used an AdaBoost algorithm to predict failure mode (FM) classification and bearing capacity of reinforced concrete columns. By comparing the AdaBoost predictions with empirical predictions, they have concluded that the AdaBoost algorithm performs better in structural design. Ben Chaabene et al. [19] have investigated machine learning techniques to accurately predict the mechanical strength of concrete. They have concluded that empirical models perform better. Marani and Nehdi [67] have used machine learning to predict the strength of PCM-integrated cementitious composites. They have observed that the gradient-boosting model gives the highest accuracy. Liu et al. [64] have studied bending curve and shear deformation analyses of reinforced concrete columns using BP and RBF neural network algorithms for seismic performance evaluation. Chakraborty et al. [23] have used machine learning models to predict the compressive behavior of high-performance concrete. They have explained the dependencies and influences of properties such as age, water-to-binder ratio, and cement with a game theory approach. Nguyen et al. [70] have compared various machine learning models for predicting concrete strengths. They have observed that the GBR and XGBoost models achieved better results. They have concluded that there has been a significant improvement in prediction accuracy and computational efficiency. Kandiri et al. [54] have used an artificial neural network method modified with different optimization algorithms to predict the strength of concrete produced using recycled aggregate. Kang et al. [55] have predicted the compressive and flexural strengths of steel fiber-reinforced concrete (SFRC) using machine learning algorithms. Ahmad et al. [5] have used machine learning algorithms to predict the compressive strength of RCA-based concrete. They have concluded that the GEP model performed better than the ANN model in predicting compressive strength. Ahmad et al. [6] have used a machine learning approach to accurately predict the properties and responses of concrete. They have observed that the ensemble model achieves better results than empirical approaches using the decision tree algorithm. Behnood and Golafshani [15] have used artificial intelligence techniques such as ANN, ANFIS, and SVR for modeling concrete behavior and mix properties. Ahmad et al. [4] have studied the compressive strength prediction of fly ash-based geopolymer concrete using decision tree, bagging regressor, and AdaBoost regressor models. They have concluded that the bagging model performed better, with an R2 value of 0.97. Thai [87] has reviewed techniques, applications, challenges, and future directions in the field of machine learning. He has reviewed studies covering applications such as structural analysis, health monitoring, design, and damage detection. Oyebisi and Alomayri [76] have used deep neural networks (DNN) to predict the strength of slag-ash-based geopolymer concrete. They have observed that DNN techniques improve the quality assurance of concrete. Almohammed and Thakur [7] have used ANN, RF, and RT models to predict the compressive strength of basalt fiber (BF) and polypropylene fiber (PPF) concrete. They have concluded that the RF model performed better than the ANN and RT models in predicting the compressive strength of concrete.

(b) **Structural Design and Analysis**

Machine learning is utilized in certain applications for the purpose of designing and analyzing structural elements.

Mohammadhassani et al. [68] used an ANFIS model to find the shear resistance of the reinforced concrete deep beams; Pal and Deswal [77] and Chou et al. [27] used an SVM model, Sanad and Saka [83] used an ANN model. For shear resistance, the ANN model has been used by Elsanadedy et al. [36] in high-strength thin beams, Abambres and Lantsoght [1] in reinforced concrete plates, Cladera and Marí [28] and El Chabib et al. [35] in reinforced concrete beams without stirrups. Moreover, Cladera and Marí [29] and El Chabib et al. [34] have used an artificial neural network model to predict the shear resistance of reinforced concrete beams made of normal-strength concrete (NSC) and high-strength concrete (HSC). Cladera and Marí [29] have used data sets from 656 reinforced concrete beams to train the artificial neural network models. Nguyen et al. [71] have used an ANN model to find the shear strength of reinforced concrete walls under seismic loads; Keshtegar et al. [58] and Parsa and Naderpour [79] have used a SVM model. Feng et al. [41] have employed an eXtreme Gradient Boosting (XGBoost) algorithm to calculate the shear strength of reinforced concrete shear walls, which are utilized as lateral load-resistant components in seismic areas. In addition, they have employed a SHapley Additive exPlanations (SHAP) method to ascertain the relative importance of the parameters that impact the shear strength estimates generated by XGBoost. Aguilar et al. [3] have studied the shear resistance of reinforced masonry walls in the earthquake zone using an ANN model.

Vanluchene and Sun [88] have solved three structural engineering problems involving a neural network for pattern recognition, a basic concrete beam design, and analysis of rectangular plates. Adeli and Park [2] have developed a counterpropagation neural network (CPN) application that includes competition and interpolation layers for the purpose of structural analysis and design. Hung and Jan [50] have developed an integrated fuzzy neural network (IFN) learning model to predict the design of concrete beams, steel structures, and the location and magnitude of the maximum bending moment in the plate. Hadi [46] has developed a neural network to optimize the design of simply supported reinforced concrete beams and fiber reinforced concrete beams in various loadings and combinations. Kaveh and Dehkordi [57] have studied analyzing, designing, and predicting the displacements of domes using a Backpropagation and a Radial Basis Functions neural network. Esteghamati and Flint [37] have used different machine learning models to predict a building's seismic sensitivity and environmental impacts. Charalampakis and Papanikolaou [25] have used an artificial neural network for the fast and accurate design of rectangular, circular, solid, and hollow reinforced concrete columns.

Papadrakakis et al. [78] have investigated the reliability analysis of complex structural systems by applying a Back Propagation algorithm to train neural networks (NN). They have related the reliability analysis to the plastic collapse limit. They have observed that the failure probability of the structural system is improved with neural networks. Caglar et al. [20] have applied a neural network formulation by training Finite Element Analysis (FEA) solutions to predict the dynamic response of both braced and unbraced plane steel frames. Lagaros and Papadrakakis [61] have used an ANN to predict the nonlinear behavior of frame systems, taking into account

earthquake effects. Wang et al. [89] used an ANN model that had been trained with data sets from experiments to guess how the load-strain relationship of concrete-filled steel tubes (CFT) would work. Kalman Šipoš et al. [52] have conducted studies to predict the seismic behavior and modeling of reinforced concrete frames with masonry wall infill using a neural network trained with data sets created from the results of 113 experimental tests. Taheri et al. [85] evaluated the prediction by using artificial intelligence (AI) and machine learning (ML) techniques for modeling and analyzing various cold-formed steel (CFS) vertical profiles with different lengths, thicknesses, and reinforcement spacing under flexural loading. They have observed that the hybrid neural network could successfully predict the normalized ultimate load.

(c) **Damage Assessment**

There are many machine learning studies used for damage detection of structural elements such as columns, beams, plates, frames, and trusses.

Jeyasehar and Sumangala [51] have developed an ANN-based algorithm for damage detection of prestressed concrete beams using vibration-based detection. Tan et al. [86] and Sadeghi et al. [82] have applied an ANN model method and vibration-based detection method for steel–concrete composite beams. Hakim et al. [47] have investigated damage detection in steel beams using an ANN algorithm using dynamic parameters to detect the location and severity of damage. Mangalathu and Jeon [66] have compared five different machine learning models (ANN, KNN, NB, RF, and DT) for the prediction of failure modes of circular reinforced concrete columns using databases tested on circular reinforced concrete columns. Naderpour et al. [69] have used ANN and DT models for failure mode detection of circular and rectangular reinforced concrete columns. Xu et al. [91] have investigated a machine learning-based approach for seismic failure mode in the corrosion of reinforced concrete columns. Davoudi et al. [31] have developed an SVM model for flexural crack detection in reinforced concrete beams and slabs based on an image-based approach with a dataset of 862 crack images from 127 samples. Davoudi et al. [32] have used an image-based approach model for shear crack detection in reinforced concrete beams and slabs with a dataset of 558 crack images obtained from 84 shear tests. Aravind et al. [8] have conducted studies for crack detection in geopolymer and conventional concrete beams using six machine learning classifications. Ye et al. [93] have developed a model called the Ci-Net using CNN algorithm for crack detection of concrete beams. They have trained their model using 762 crack images.

For the detection of cracks in slab and plate-type structural elements, classification methods for image-level cracks [30, 33, 44, 81] and segmentation techniques for pixel-level cracks [11, 63, 74, 92]. Cha et al. [21] have developed a CNN model for surface crack detection in concrete structures using datasets of 332 cracked concrete images. Cha et al. [22] have developed CNN models for different types of damage, such as concrete cracks, bolt corrosion, steel delamination, and steel corrosion. Nguyen et al. [73] and Bang et al. [11] have used CNN models for road crack detection, Dung et al. [33] for crack detection in gusset plate welded joints of steel bridges, and Dais et al. [30] for crack detection in masonry structures.

Wu et al. [90] have used an ANN model for damage detection of reinforced concrete frame structures under earthquake motions. Huang and Burton [49] have compared six different machine learning models and predicted failure modes on reinforced concrete frames with infills. Szewczyk and Hajela [84] and Pillai and Krishnapillai [80] have used an ANN model for damage detection of frame and truss systems. Kao and Hung [56] and Beheshti Aval et al. [13] have developed an ANN model and investigated vibration-based damage detection of frame structures. Chang et al. [24] have developed an ANN model for damage detection in a three-story shear building model. Machavaram and Shankar [65] have performed damage detection for both steel and truss systems using an improved RBF neural networks model. Kim et al. [60] have studied damage detection of steel frame structures using the deep convolutional neural network-based damage locating (DCNN-DL) method.

(d) **Control Systems**

Ocak et al. [75] have used machine learning classification algorithms to predict the damping capacity of seismic base isolators with a dataset of SDOF structures under remote fault earthquakes. They have concluded that the machine learning model predicts the damping of isolators with an accuracy of over 96%. Chong et al. [26] have used an SVM model to predict the damage states of magnetorheological dampers proposed for smart structures with nonlinear characteristics. Khalid et al. [59] have predicted the response of a new magneto-rheological damper using a dynamic neural network model. They have used a Bouc-Wen model to generate data sets to be used in training the model. Bae et al. [9] have used machine learning for fatigue prediction of a root-shaped metallic damper with plastic hinges for earthquake resistance. Nguyen et al. [72] have used Linear, Ridge, Lasso, ANN, SVM, and random forest methods to predict the maximum displacement of a seismic isolation system calculated using different earthquake scenarios. Yucel et al. [96] have compared ANN-based formulation and classical formulation for the optimization of TMD parameters. They have observed that the ANN-based formulation works more effectively than the classical formulation. They have concluded that the ANN-based formulation is effective in reducing the maximum reaction. Yucel et al. [95] have used the ANN model found by reducing the maximum structural responses in single degree of freedom (SDOF) and multi-degree of freedom (MDOF) structures with TMD. They concluded that it is more effective than the optimum results found with the existing formulation. Barakat [12] has investigated ANN models for the design of seismic isolator systems. He has observed that ANN models are a powerful tool for design. Li et al. [62] have used an SVM-based semi-active control model to measure the responses of magnetorheological dampers used for seismic protection, such as displacement, velocity, and acceleration. They have observed that the results of the SVM-based semi-active control model performed better than the results of magnetorheological dampers. Bahiuddin et al. [10] have used an ELM tribas activation function to predict the nonlinear behavior of magnetorheological dampers. They have compared it with an ANN for training time and accuracy. They have concluded that the ELM is faster than the ANN. Farrokhi and Rahimi [38] have used an SVM model to investigate the behavior of steel frame systems with a mass damper. Habib and Yildirim [45] have

applied the physics-informed neural network model for the fast design of quintuple friction pendulum isolators using a dataset of 35,000 isolators.

3 Conclusion

As seen in the presented reviews, the use of data analysis and machine learning in civil engineering studies has begun to be applied. Utilizing these methods proves to be efficient in resolving structural issues. Machine learning methods are applied to find optimal solutions quickly and with high accuracy in complex and high-throughput design domains. In this context, the application of data analysis and machine learning in civil engineering involves the development of capabilities to explore different analysis methods, decrease long analysis times, and handle multiple problems together. This improbable and adaptable approach enables more efficient exploration of the extensive range of studies in civil engineering and the investigation of optimal designs.

In the future, machine-learning approaches are likely to become increasingly prevalent in civil engineering. Prediction methods using machine learning for a realistic and detailed analysis can solve many uncertain problems. This solution enables the attainment of a more efficient method through constant self-improvement and ensures an optimal approach.

References

1. Abambres, M., Lantsoght, E.O.L.: Neural network-based formula for shear capacity prediction of one-way slabs under concentrated loads. Eng. Struct. **211**, 110501 (2020). https://doi.org/10.1016/j.engstruct.2020.110501
2. Adeli, H., Park, H.S.: Counterpropagation neural networks in structural engineering. J. Struct. Eng. (U.S.) **121**, 1205–1212 (1995). https://doi.org/10.1061/(ASCE)0733-9445(1995)121:8(1205)
3. Aguilar, V., Sandoval, C., Adam, J.M., Garzón-Roca, J., Valdebenito, G.: Prediction of the shear strength of reinforced masonry walls using a large experimental database and artificial neural networks. Struct. Infrastruct. Eng. **12**, 1661–1674 (2016). https://doi.org/10.1080/15732479.2016.1157824
4. Ahmad, A., Ahmad, W., Aslam, F., Joyklad, P.: Compressive strength prediction of fly ash-based geopolymer concrete via advanced machine learning techniques. Case Stud. Construct. Mater. **16**, e00840 (2022). https://doi.org/10.1016/j.cscm.2021.e00840
5. Ahmad, A., Chaiyasarn, K., Farooq, F., Ahmad, W., Suparp, S., Aslam, F.: Compressive strength prediction via gene Expression Programming (GEP) and Artificial Neural Network (ANN) for concrete containing RCA. Buildings **11**, 324 (2021). https://doi.org/10.3390/buildings11080324
6. Ahmad, A., Farooq, F., Niewiadomski, P., Ostrowski, K., Akbar, A., Aslam, F., Alyousef, R.: Prediction of compressive strength of fly ash based concrete using individual and ensemble algorithm. Materials **14**, 794 (2021). https://doi.org/10.3390/ma14040794

7. Almohammed, F., Thakur, M.S.: Forecasting compressive strength of concrete with basalt and polypropylene fiber by using ANN, RF and RT models. Asian J. Civ. Eng. **25**, 1671–1690 (2024). https://doi.org/10.1007/s42107-023-00870-4
8. Aravind, N., Nagajothi, S., Elavenil, S.: Machine learning model for predicting the crack detection and pattern recognition of geopolymer concrete beams. Constr. Build. Mater. **297**, 123785 (2021). https://doi.org/10.1016/j.conbuildmat.2021.123785
9. Bae, J., Lee, C.-H., Park, M., Alemayehu, R.W., Ryu, J., Ju, Y.K.: Modified low-cycle fatigue estimation using machine learning for radius-cut coke-shaped metallic damper subjected to cyclic loading. Int. J. Steel Struct. **20**, 1849–1858 (2020). https://doi.org/10.1007/s13296-020-00377-7
10. Bahiuddin, I., Imaduddin, F., Mazlan, S.A., Ariff, Mohd.H.M., Mohmad, K.B., Ubaidillah, Choi, S.-B.: Accurate and fast estimation for field-dependent nonlinear damping force of meandering valve-based magnetorheological damper using extreme learning machine method. Sens. Actuators A: Phys. **318**, 112479 (2021). https://doi.org/10.1016/j.sna.2020.112479
11. Bang, S., Park, S., Kim, H., Kim, H.: Encoder–decoder network for pixel-level road crack detection in black-box images. Comput.-Aided Civil Infrastruct. Eng. **34**, 713–727 (2019). https://doi.org/10.1111/mice.12440
12. Barakat, S.:Design of the base isolation system with artificial neural network models. In: Proceedings of the 2020 4th International Conference on Compute and Data Analysis, ICCDA'20, pp. 79–83. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3388142.3388169
13. Beheshti Aval, S.B., Ahmadian, V., Maldar, M., Darvishan, E.: Damage detection of structures using signal processing and artificial neural networks. Adv. Struct. Eng. **23**, 884–897 (2020). https://doi.org/10.1177/1369433219886079
14. Behnood, A., Behnood, V., Modiri Gharehveran, M., Alyamac, K.E.: Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm. Constr. Build. Mater. **142**, 199–207 (2017). https://doi.org/10.1016/j.conbuildmat.2017.03.061
15. Behnood, A., Golafshani, E.M.: Artificial intelligence to model the performance of concrete mixtures and elements: a review. Arch. Comput. Methods Eng. **29**, 1941–1964 (2022). https://doi.org/10.1007/s11831-021-09644-0
16. Behnood, A., Golafshani, E.M.: Machine learning study of the mechanical properties of concretes containing waste foundry sand. Constr. Build. Mater. **243**, 118152 (2020). https://doi.org/10.1016/j.conbuildmat.2020.118152
17. Behnood, A., Golafshani, E.M.: Predicting the compressive strength of silica fume concrete using hybrid artificial neural network with multi-objective grey wolves. J. Clean. Prod. **202**, 54–64 (2018). https://doi.org/10.1016/j.jclepro.2018.08.065
18. Behnood, A., Verian, K.P., Modiri Gharehveran, M.: Evaluation of the splitting tensile strength in plain and steel fiber-reinforced concrete based on the compressive strength. Constr. Build. Mater. **98**, 519–529 (2015). https://doi.org/10.1016/j.conbuildmat.2015.08.124
19. Ben Chaabene, W., Flah, M., Nehdi, M.L.: Machine learning prediction of mechanical properties of concrete: critical review. Constr. Build. Mater. **260**, 119889 (2020). https://doi.org/10.1016/j.conbuildmat.2020.119889
20. Caglar, N., Pala, M., Elmas, M., Mercan Eryılmaz, D.: A new approach to determine the base shear of steel frame structures. J. Constr. Steel Res. **65**, 188–195 (2009). https://doi.org/10.1016/j.jcsr.2008.07.012
21. Cha, Y.-J., Choi, W., Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. Comput.-Aided Civil Infrastruct. Eng. **32**, 361–378 (2017). https://doi.org/10.1111/mice.12263
22. Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., Büyüköztürk, O.: Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput.-Aided Civil Infrastruct. Eng. **33**, 731–747 (2018). https://doi.org/10.1111/mice.12334
23. Chakraborty, D., Awolusi, I., Gutierrez, L.: An explainable machine learning model to predict and elucidate the compressive behavior of high-performance concrete. Res. Eng. **11**, 100245 (2021). https://doi.org/10.1016/j.rineng.2021.100245

24. Chang, M., Kim, J.K., Lee, J.: Hierarchical neural network for damage detection using modal parameters. Struct. Eng. Mech. **70**, 457–466 (2019). https://doi.org/10.12989/sem.2019.70.4.457
25. Charalampakis, A.E., Papanikolaou, V.K.: Machine learning design of R/C columns. Eng. Struct. **226**, 111412 (2021). https://doi.org/10.1016/j.engstruct.2020.111412
26. Chong, J.W., Kim, Y., Chon, K.H.: Nonlinear multiclass support vector machine–based health monitoring system for buildings employing magnetorheological dampers. J. Intell. Mater. Syst. Struct. **25**, 1456–1468 (2014). https://doi.org/10.1177/1045389X13507343
27. Chou, J.-S., Ngo, N.-T., Pham, A.-D.: Shear strength prediction in reinforced concrete deep beams using nature-inspired metaheuristic support vector regression. J. Comput. Civ. Eng. **30**, 04015002 (2016). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000466
28. Cladera, A., Marí, A.R.: Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part I: beams without stirrups. Eng. Struct. **26**, 917–926 (2004). https://doi.org/10.1016/j.engstruct.2004.02.010
29. Cladera, A., Marí, A.R.: Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part II: beams with stirrups. Eng. Struct. **26**, 927–936 (2004). https://doi.org/10.1016/j.engstruct.2004.02.011
30. Dais, D., Bal, İE., Smyrou, E., Sarhosis, V.: Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. Autom. Constr. **125**, 103606 (2021). https://doi.org/10.1016/j.autcon.2021.103606
31. Davoudi, R., Miller, G.R., Kutz, J.N.: Data-driven vision-based inspection for reinforced concrete beams and slabs: quantitative damage and load estimation. Autom. Constr. **96**, 292–309 (2018). https://doi.org/10.1016/j.autcon.2018.09.024
32. Davoudi, R., Miller, G.R., Kutz, J.N.: Structural load estimation using machine vision and surface crack patterns for shear-critical RC beams and slabs. J. Comput. Civil Eng. **32** (2018). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000766
33. Dung, C.V., Sekiya, H., Hirano, S., Okatani, T., Miki, C.: A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks. Autom. Constr. **102**, 217–229 (2019). https://doi.org/10.1016/j.autcon.2019.02.013
34. El Chabib, H., Nehdi, M., Saïd, A.: Predicting the effect of stirrups on shear strength of reinforced normal-strength concrete (NSC) and high-strength concrete (HSC) slender beams using artificial intelligence. Can. J. Civ. Eng. **33**, 933–944 (2006). https://doi.org/10.1139/L06-033
35. El-Chabib, H., Nehdi, M., Said, A.: Predicting shear capacity of NSC and HSC slender beams without stirrups using artificial intelligence. Comput. Concrete **2**, 79–96 (2005). https://doi.org/10.12989/cac.2005.2.1.079
36. Elsanadedy, H.M., Abbas, H., Al-Salloum, Y.A., Almusallam, T.H.: Shear strength prediction of HSC slender beams without web reinforcement. Mater. Struct. **49**, 3749–3772 (2016). https://doi.org/10.1617/s11527-015-0752-x
37. Esteghamati, M.Z., Flint, M.M.: Developing data-driven surrogate models for holistic performance-based assessment of mid-rise RC frame buildings at early design. Eng. Struct. **245**, 112971 (2021). https://doi.org/10.1016/j.engstruct.2021.112971
38. Farrokhi, F., Rahimi, S.: Supervised probabilistic failure prediction of tuned mass damper-equipped high steel frames using machine learning methods. Studia Geotechnica et Mechanica **42**, 179–190 (2020). https://doi.org/10.2478/sgem-2019-0043
39. Feng, D.-C., Liu, Z.-T., Wang, X.-D., Chen, Y., Chang, J.-Q., Wei, D.-F., Jiang, Z.-M.: Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach. Constr. Build. Mater. **230**, 117000 (2020). https://doi.org/10.1016/j.conbuildmat.2019.117000
40. Feng, D.-C., Liu, Z.-T., Wang, X.-D., Jiang, Z.-M., Liang, S.-X.: Failure mode classification and bearing capacity prediction for reinforced concrete columns based on ensemble machine learning algorithm. Adv. Eng. Inform. **45**, 101126 (2020). https://doi.org/10.1016/j.aei.2020.101126
41. Feng, D.-C., Wang, W.-J., Mangalathu, S., Taciroglu, E.: Interpretable XGBoost-SHAP machine-learning model for shear strength prediction of squat RC walls. J. Struct. Eng. (U.S.) **147** (2021). https://doi.org/10.1061/(ASCE)ST.1943-541X.0003115

42. Gholampour, A., Mansouri, I., Kisi, O., Ozbakkaloglu, T.: Evaluation of mechanical properties of concretes containing coarse recycled concrete aggregates using multivariate adaptive regression splines (MARS), M5 model tree (M5Tree), and least squares support vector regression (LSSVR) models. Neural Comput. Appl. **32**, 295–308 (2020). https://doi.org/10.1007/s00521-018-3630-y
43. Golafshani, E.M., Behnood, A., Arashpour, M.: Predicting the compressive strength of normal and high-performance concretes using ANN and ANFIS hybridized with Grey Wolf Optimizer. Constr. Build. Mater. **232**, 117266 (2020). https://doi.org/10.1016/j.conbuildmat.2019.117266
44. Gulgec, N.S., Takáč, M., Pakzad, S.N.: Convolutional neural network approach for robust structural damage detection and localization. J. Comput. Civil Eng. **33** (2019). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000820
45. Habib, A., Yildirim, U.: Developing a physics-informed and physics-penalized neural network model for preliminary design of multi-stage friction pendulum bearings. Eng. Appl. Artif. Intell. **113**, 104953 (2022). https://doi.org/10.1016/j.engappai.2022.104953
46. Hadi, M.N.S.: Neural networks applications in concrete structures. Comput. Struct. **81**, 373–381 (2003). https://doi.org/10.1016/S0045-7949(02)00451-0
47. Hakim, S.J.S., Razak, H.A., Ravanfar, S.A., Mohammadhassani, M.: Structural damage detection using soft computing method. In: Wicks, A. (ed.) Structural Health Monitoring, vol. 5, pp. 143–151. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-04570-2_16
48. Han, T., Siddique, A., Khayat, K., Huang, J., Kumar, A.: An ensemble machine learning approach for prediction and optimization of modulus of elasticity of recycled aggregate concrete. Constr. Build. Mater. **244**, 118271 (2020). https://doi.org/10.1016/j.conbuildmat.2020.118271
49. Huang, H., Burton, H.V.: Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning. J. Build. Eng. **25** (2019). https://doi.org/10.1016/j.jobe.2019.100767
50. Hung, S.-L., Jan, J.C.: Machine learning in engineering analysis and design: an integrated fuzzy neural network learning model. Comput.-Aided Civil Infrastruct. Eng. **14**, 207–219 (1999). https://doi.org/10.1111/0885-9507.00142
51. Jeyasehar, C.A., Sumangala, K.: Damage assessment of prestressed concrete beams using artificial neural network (ANN) approach. Comput. Struct. **84**, 1709–1718 (2006). https://doi.org/10.1016/j.compstruc.2006.03.005
52. Kalman Šipoš, T., Sigmund, V., Hadzima-Nyarko, M.: Earthquake performance of infilled frames using neural networks and experimental database. Eng. Struct. **51**, 113–127 (2013). https://doi.org/10.1016/j.engstruct.2012.12.038
53. Kandiri, A., Mohammadi Golafshani, E., Behnood, A.: Estimation of the compressive strength of concretes containing ground granulated blast furnace slag using hybridized multi-objective ANN and salp swarm algorithm. Constr. Build. Mater. **248**, 118676 (2020). https://doi.org/10.1016/j.conbuildmat.2020.118676
54. Kandiri, A., Sartipi, F., Kioumarsi, M.: Predicting compressive strength of concrete containing recycled aggregate using modified ANN with different optimization algorithms. Appl. Sci. **11**, 485 (2021). https://doi.org/10.3390/app11020485
55. Kang, M.-C., Yoo, D.-Y., Gupta, R.: Machine learning-based prediction for compressive and flexural strengths of steel fiber-reinforced concrete. Constr. Build. Mater. **266**, 121117 (2021). https://doi.org/10.1016/j.conbuildmat.2020.121117
56. Kao, C.Y., Hung, S.-L.: Detection of structural damage via free vibration responses generated by approximating artificial neural networks. Comput. Struct. **81**, 2631–2644 (2003). https://doi.org/10.1016/S0045-7949(03)00323-7
57. Kaveh, A., Dehkordi, M.R.: Neural networks for the analysis and design of domes. Int. J. Space Struct. **18**, 181–193 (2003). https://doi.org/10.1260/026635103322437463
58. Keshtegar, B., Nehdi, M.L., Trung, N.-T., Kolahchi, R.: Predicting load capacity of shear walls using SVR–RSM model. Appl. Soft Comput. **112**, 107739 (2021). https://doi.org/10.1016/j.asoc.2021.107739

59. Khalid, M., Yusof, R., Joshani, M., Selamat, H., Joshani, M.: Nonlinear identification of a magneto-rheological damper based on dynamic neural networks. Comput.-Aided Civil Infrastruct. Eng. **29**, 221–233 (2014). https://doi.org/10.1111/mice.12005
60. Kim, B., Yuvaraj, N., Park, H.W., Preethaa, K.R.S., Pandian, R.A., Lee, D.-E.: Investigation of steel frame damage based on computer vision and deep learning. Autom. Constr. **132**, 103941 (2021). https://doi.org/10.1016/j.autcon.2021.103941
61. Lagaros, N.D., Papadrakakis, M.: Neural network based prediction schemes of the non-linear seismic response of 3D buildings. Adv. Eng. Softw. Civil-Comp. **44**, 92–115 (2012). https://doi.org/10.1016/j.advengsoft.2011.05.033
62. Li, C., Liu, Q., Lan, S.: Application of support vector machine-based semiactive control for seismic protection of structures with magnetorheological dampers. Math. Probl. Eng. **2012**, e268938 (2012). https://doi.org/10.1155/2012/268938
63. Li, S., Zhao, X., Zhou, G.: Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. Comput.-Aided Civil Infrastruct. Eng. **34**, 616–634 (2019). https://doi.org/10.1111/mice.12433
64. Liu, Q., Sun, P., Fu, X., Zhang, J., Yang, H., Gao, H., Li, Y.: Comparative analysis of BP neural network and RBF neural network in seismic performance evaluation of pier columns. Mech. Syst. Signal Process. **141**, 106707 (2020). https://doi.org/10.1016/j.ymssp.2020.106707
65. Machavaram, R., Shankar, K.: Structural damage identification using improved RBF neural networks in frequency domain. Adv. Struct. Eng. **15**, 1689–1703 (2012). https://doi.org/10.1260/1369-4332.15.10.1689
66. Mangalathu, S., Jeon, J.-S.: Machine learning-based failure mode recognition of circular reinforced concrete bridge columns: comparative study. J. Struct. Eng. (U.S.) **145** (2019). https://doi.org/10.1061/(ASCE)ST.1943-541X.0002402
67. Marani, A., Nehdi, M.L.: Machine learning prediction of compressive strength for phase change materials integrated cementitious composites. Constr. Build. Mater. **265**, 120286 (2020). https://doi.org/10.1016/j.conbuildmat.2020.120286
68. Mohammadhassani, M., Saleh, A.M., Suhatril, M., Safa, M.: Fuzzy modelling approach for shear strength prediction of RC deep beams. Smart Struct. Syst. **16**, 497–519 (2015). https://doi.org/10.12989/sss.2015.16.3.497
69. Naderpour, H., Mirrashid, M., Parsa, P.: Failure mode prediction of reinforced concrete columns using machine learning methods. Eng. Struct. **248**, 113263 (2021). https://doi.org/10.1016/j.engstruct.2021.113263
70. Nguyen, D.-D., Tran, V.-L., Ha, D.-H., Nguyen, V.-Q., Lee, T.-H.: A machine learning-based formulation for predicting shear capacity of squat flanged RC walls. Structures **29**, 1734–1747 (2021). https://doi.org/10.1016/j.istruc.2020.12.054
71. Nguyen, H., Vu, T., Vo, T.P., Thai, H.-T.: Efficient machine learning models for prediction of concrete strengths. Constr. Build. Mater. **266**, 120950 (2021). https://doi.org/10.1016/j.conbuildmat.2020.120950
72. Nguyen, H.D., Dao, N.D., Shin, M.: Machine learning-based prediction for maximum displacement of seismic isolation systems. J. Build. Eng. **51**, 104251 (2022). https://doi.org/10.1016/j.jobe.2022.104251
73. Nguyen, N.H.T., Perry, S., Bone, D., Le, H.T., Nguyen, T.T.: Two-stage convolutional neural network for road crack detection and segmentation. Expert Syst. Appl. **186**, 115718 (2021). https://doi.org/10.1016/j.eswa.2021.115718
74. Ni, F., Zhang, J., Chen, Z.: Pixel-level crack delineation in images with convolutional feature fusion. Struct. Control. Health Monit. **26**, e2286 (2019). https://doi.org/10.1002/stc.2286
75. Ocak, A., Işıkdağ, Ü., Bekdaş, G., Nigdeli, S., Kim, S., Geem, Z.: Prediction of damping capacity demand in seismic base isolators via machine learning. CMES **138**, 2899–2924 (2023). https://doi.org/10.32604/cmes.2023.030418
76. Oyebisi, S., Alomayri, T.: Artificial intelligence-based prediction of strengths of slag-ash-based geopolymer concrete using deep neural networks. Constr. Build. Mater. **400**, 132606 (2023). https://doi.org/10.1016/j.conbuildmat.2023.132606

77. Pal, M., Deswal, S.: Support vector regression based shear strength modelling of deep beams. Comput. Struct. **89**, 1430–1439 (2011). https://doi.org/10.1016/j.compstruc.2011.03.005
78. Papadrakakis, M., Papadopoulos, V., Lagaros, N.D.: Structural reliability analyis of elastic-plastic structures using neural networks and Monte Carlo simulation. Comput. Methods Appl. Mech. Eng. **136**, 145–163 (1996). https://doi.org/10.1016/0045-7825(96)01011-0
79. Parsa, P., Naderpour, H.: Shear strength estimation of reinforced concrete walls using support vector regression improved by teaching–learning-based optimization, particle Swarm optimization, and Harris Hawks Optimization algorithms. J. Build. Eng. **44**, 102593 (2021). https://doi.org/10.1016/j.jobe.2021.102593
80. Pillai, P., Krishnapillai, S.: A hybrid neural network strategy for identification of structural parameters. Struct. Infrastruct. Eng. **6**, 379–391 (2010). https://doi.org/10.1080/15732470701718197
81. Rao, A.S., Nguyen, T., Palaniswami, M., Ngo, T.: Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure. Struct. Health Monit. **20**, 2124–2142 (2021). https://doi.org/10.1177/1475921720965445
82. Sadeghi, F., Yu, Y., Zhu, X., Li, J.: Damage identification of steel-concrete composite beams based on modal strain energy changes through general regression neural network. Eng. Struct. **244**, 112824 (2021). https://doi.org/10.1016/j.engstruct.2021.112824
83. Sanad, A., Saka, M.P.: Prediction of ultimate shear strength of reinforced-concrete deep beams using neural networks. J. Struct. Eng. **127**, 818–828 (2001). https://doi.org/10.1061/(ASCE)0733-9445(2001)127:7(818)
84. Szewczyk, Z.P., Hajela, P.: Damage detection in structures based on feature-sensitive neural networks. J. Comput. Civ. Eng. **8**, 163–178 (1994). https://doi.org/10.1061/(ASCE)0887-3801(1994)8:2(163)
85. Taheri, E., Esgandarzadeh Fard, S., Zandi, Y., Samali, B.: Experimental and numerical investigation of an innovative method for strengthening cold-formed steel profiles in bending throughout finite element modeling and application of neural network based on feature selection method. Appl. Sci. **11**, 5242 (2021). https://doi.org/10.3390/app11115242
86. Tan, Z.X., Thambiratnam, D.P., Chan, T.H.T., Gordan, M., Abdul Razak, H.: Damage detection in steel-concrete composite bridge using vibration characteristics and artificial neural network. Struct. Infrastruct. Eng. **16**, 1247–1261 (2020). https://doi.org/10.1080/15732479.2019.1696378
87. Thai, H.-T.: Machine learning for structural engineering: a state-of-the-art review. Structures **38**, 448–491 (2022). https://doi.org/10.1016/j.istruc.2022.02.003
88. Vanluchene, R.D., Sun, R.: Neural networks in structural engineering. Comput.-Aided Civil Infrastruct. Eng. **5**, 207–215 (1990). https://doi.org/10.1111/j.1467-8667.1990.tb00377.x
89. Wang, Y., Liu, Z.Q., Zhang, M.: Prediction of mechanical behavior of concrete filled steel tube structure using artificial neural network. Appl. Mech. Mater. **368–370**, 1095–1098 (2013). https://doi.org/10.4028/www.scientific.net/AMM.368-370.1095
90. Wu, X., Ghaboussi, J., Garrett, J.H.: Use of neural networks in detection of structural damage. Comput. Struct. **42**, 649–659 (1992). https://doi.org/10.1016/0045-7949(92)90132-J
91. Xu, J.-G., Hong, W., Zhang, J., Hou, S.-T., Wu, G.: Seismic performance assessment of corroded RC columns based on data-driven machine-learning approach. Eng. Struct. **255**, 113936 (2022). https://doi.org/10.1016/j.engstruct.2022.113936
92. Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., Yang, X.: Automatic pixel-level crack detection and measurement using fully convolutional network. Comput.-Aided Civil Infrastruct. Eng. **33**, 1090–1109 (2018). https://doi.org/10.1111/mice.12412
93. Ye, X.-W., Jin, T., Chen, P.-Y.: Structural crack detection using deep learning–based fully convolutional networks. Adv. Struct. Eng. **22**, 3412–3419 (2019). https://doi.org/10.1177/1369433219836292
94. Young, B.A., Hall, A., Pilon, L., Gupta, P., Sant, G.: Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: new insights from statistical analysis and machine learning methods. Cem. Concr. Res. **115**, 379–388 (2019). https://doi.org/10.1016/j.cemconres.2018.09.006

95. Yucel, M., Bekdaş, G., Nigdeli, S.M., Sevgen, S.: Estimation of optimum tuned mass damper parameters via machine learning. J. Build. Eng. **26**, 100847 (2019). https://doi.org/10.1016/j.jobe.2019.100847
96. Yucel, M., Nigdeli, S.M., Bekdaş, G.: Evaluation of artificial neural network-based formulations for tuned mass dampers. CJSMEC **7**, 17 (2021). https://doi.org/10.20528/cjsmec.2021.01.003

Application of Artificial Intelligence (AI) in Civil Engineering

Temitope Funmilayo Awolusi, Bernard Chukwuemeka Finbarrs-Ezema, Isaac Munachimdinamma Chukwudulue, and Marc Azab

Abstract Hard computing generally deals with precise data, which provides ideal solutions to problems. However, in the civil engineering field, amongst other disciplines, that is not always the case as real-world systems are continuously changing. Here lies the need to explore soft computing methods and artificial intelligence to solve civil engineering shortcomings. The integration of advanced computational models, including Artificial Neural Networks (ANNs), Fuzzy Logic, Genetic Algorithms (GAs), and Probabilistic Reasoning, has revolutionized the domain of civil engineering. These models have significantly advanced diverse sub-fields by offering innovative solutions and improved analysis capabilities. Sub-fields such as: slope stability analysis, bearing capacity, water quality and treatment, transportation systems, air quality, structural materials, etc. ANNs predict non-linearities and provide accurate estimates. Fuzzy logic uses an efficient decision-making process to provide a more precise assessment of systems. Lastly, while GAs optimizes models (based on evolutionary processes) for better outcomes, probabilistic reasoning lowers their statistical uncertainties.

Keywords Artificial intelligence · Artificial neural networks · Fuzzy logic · Genetic algorithms · Probabilistic reasoning

T. F. Awolusi (✉) · B. C. Finbarrs-Ezema
Department of Civil Engineering, Afe Babalola University, Ado Ekiti, Ekiti State, Nigeria
e-mail: awolusitf@abuad.edu.ng

B. C. Finbarrs-Ezema
e-mail: finbarrs-ezemabc@abuad.edu.ng

I. M. Chukwudulue
Department of Electrical and Electronics Engineering, Afe Babalola University, Ado Ekiti, Ekiti State, Nigeria

M. Azab
Department of Civil Engineering, American University of the Middle East, Egaila, Kuwait
e-mail: marc.azab@aum.edu.kw

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_2

1 Introduction

The globe transformation drive from analogue to artificial intelligence systems of operation has necessitated the application of Artificial Intelligence (AI) in Civil Engineering. The profession can broadly be divided into four major aspects: Structural and construction Engineering, Transportation Engineering, Geo-technical and Water Resources, and Environmental Engineering. The process involves intelligence-based design and methodologies that rely on data for execution and implementation. The application of AI in civil Engineering involves combining advanced machine learning techniques to achieve improved construction practice, promoting sustainability in the construction industry, and improving infrastructural design in the built environment. Several machine learning techniques and algorithms, like artificial neural network, deep learning, fuzzy logic, etc., have been applied in estimating different properties such as concrete composites, slope failure susceptibility, and soil compaction parameters.

2 Applications in Geotechnical Engineering

2.1 Slope Stability Analysis

Artificial neural networks (ANN) can be trained to accurately predict three-dimensional slope stability [1]. Novel models have been used to estimate earth slope stability based on ANN with similar accuracies as the existing mathematical expressions [2]. Qian et al. [3] adopted ANN to develop a stability assessment tool for in-homogeneous soil slopes to minimize manual reading errors when using slope stability chart solutions. ANN are often used in conjunction with other methods to predict the factor of safety (FOS) of slopes [4]. The influence of slope engineering parameters on their stability was investigated using ANN [5].

By proper training, an ANN, as illustrated in Fig. 1, with a desirable transfer function and a suitable number of hidden layers, can reasonably predict the nonlinearities and provide an accurate slope stability estimation.

Using slope tangent, angle of internal friction, slope height, cohesion, unit weight, and the stability number as input parameters, Abdalla et al. [7] concluded that ANN models are reliable, simple, and valid computational tools for predicting the FOS and for assessing the stability of slopes of clayey soil. Training and testing data-sets typically used for the ANN slope stability analysis are obtained using various limit equilibrium analysis methods [8–11].

Many factors contribute to the considerable uncertainty surrounding the stability of slopes, and considering these uncertainties, designees have been forced to include a safety factor [12]. The theory of fuzzy sets deals with the uncertain nature of soil parameters and uncertainties involved in the analysis [13]. Fuzzy set theory is a

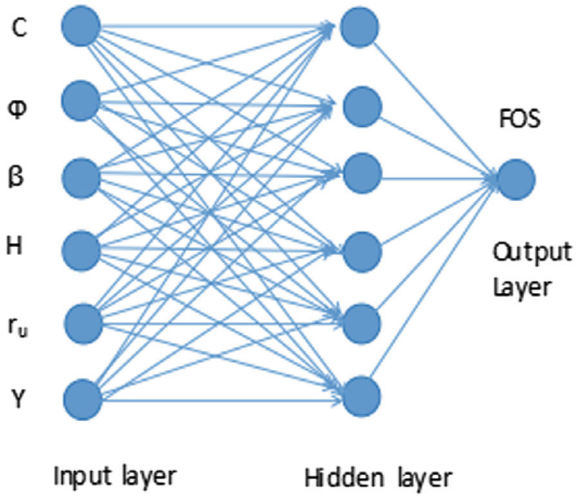


Fig. 1 Estimation of slope stability using ANN [6]

valuable tool for rock engineers and engineering geologists who study rock slope stability [14].

Mohamed et al. [15] set out to use a fuzzy logic system to predict the stability of slopes. Several significant parameters were employed as inputs, including the internal friction angle, slope height, unit weight of slope material, slope angle, coefficient of cohesion, and internal friction angle; the output parameter was the safety factor. The outcomes demonstrated its high accuracy in predicting the safety factors. Fuzzy logic-based decision-making was used by Azarafza et al. [16] to quickly evaluate block-toppling failure instability in discontinuous rock slopes, as supported by kinematic analysis applied to actual cases. They made the point that fuzzy logic enhances kinematic analyses by providing a more precise stability assessment.

Genetic algorithm (GA), due to its efficiency and stability in finding global optimum solutions, was used to improve spatial landslide prediction by Palazzolo et al. [17]. Genetic algorithms are usually hybridized with other methods for the calculation of FOS of slopes [18–20]. GA can be used to quickly find the non-circular failure surface with the lowest safety factor, as failure surfaces tend to be non-circular for layered slopes [21].

In order to examine the convergence of the failure probability of soil slopes reinforced with piles, Chen et al. [22] proposed a method for determining the minimum samples of simulation iterations. The findings showed that boring piles would create multiple potential sliding surfaces, increasing the uncertainty of slope failures. In Ji et al. [23] study, where the seismic stability of earth slopes subjected to deep failure was probabilistically investigated, a first-order reliability method was introduced for addressing uncertainties and computing the probability of failure. Chen et al. [24] employed a probabilistic analysis method to assess the failure probability of submarine landslides caused by hydrate exploitation. According to probabilistic analyses,

the failure probability was significantly influenced by both the slope angle and the production temperature.

Probability analysis on slope stability includes the combination of random field theory and the conventional slope stability evaluation methods, such as the random limit equilibrium method (RLEM) and the random finite element method (RFEM) [25].

2.2 *Soil Behavior*

Rani's [26] proposed ANN model was used to predict soil engineering properties like permeability, compressibility, and shear strength parameters, and was found to be quite satisfactory. Similarly, Kiran and Lal [27] adopted ANN as a prediction tool to determine the internal friction angle and cohesion based on some index properties such as water content, plasticity index, bulk density, sand %, silt %, and clay %. In the same study, a simplified methodology for ANN is presented, as shown in Fig. 2.

ANN is used in Rashidianand and Hassanlourad [28] work to describe the complex behavior of different carbonate soils accurately. In order to validate a deep learning method proposed by Zhang et al. [29] for modeling soil's stress-strain behavior, ANN was used as a benchmark.

In Dewidar et al. [30] study, a fuzzy-based model that is more accurate than conventional methods was developed to predict soil infiltration rate. The developed fuzzy model was tested against the observed data and multiple linear regression.

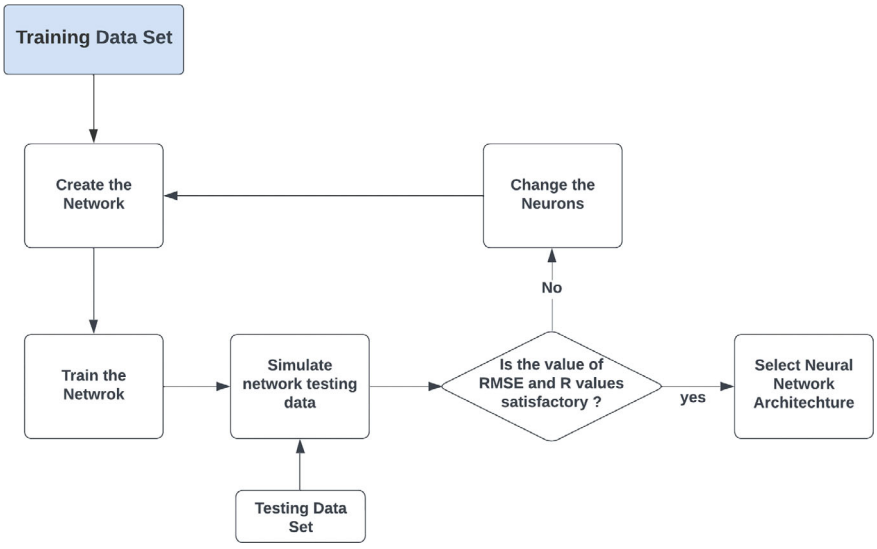


Fig. 2 Simplified methodology for ANN [27]

The coefficient of determination, root mean square error, mean absolute error, model efficiency, and overall index of the fuzzy model were 0.953, 1.53, 1.28, 0.953, and 0.954, respectively. The corresponding MLR model values were 0.913, 2.37, 1.92, 0.913, and 0.914, respectively. The results also showed that clay is the most important factor in predicting soil infiltration rate, as silt and clay were used as input variables. Sujatha et al. [31] proposed a fuzzy knowledge-based model that can be accurately and quickly used to predict the soil type and its rating for suitability in airfield applications without a need to perform any manual work, such as using tables or charts. Fişne et al. [32] found that peak particle velocity (PPV) can be used to assess ground vibration levels. The distance between the blast face and the vibration monitoring point and the charge weight per delay are employed as fuzzy logic parameters to estimate the PPV. The comparison of the measured and projected values of PPV revealed that the fuzzy model's correlation coefficient (0.96) is greater than the regression model's (0.82). Moonjun et al. [33] fine-scale soil mapping research revealed that fuzzy logic may be used to map with a relatively low density of soil samples, which may reduce costs and inconsistencies compared to standard methods.

The results of a GA-based neural network by Johari et al. [34] were compared with experimental results and were found to be accurate and robust in modelling the mechanical behaviour of unsaturated soils. Hassanlourad et al. [35] inculcated GA in an attempt to predict the compaction parameters of soils by ANN. They trained and tested their model with 212 data-sets and with the aid of sensitivity analysis, found that liquid limit and plastic limit were the most influential parameters on the compaction parameters. GA models were used to predict the unconfined compressive strength of geopolymer stabilized clayey soils [36]. Based on a comprehensive set of triaxial tests data, GA-based ANN was concluded to have a good potential in predicting the shear behavior of carbonate sands [37]. GA also performs well in solving soil-structure interaction problems [38].

A Bayesian framework for probabilistic soil stratification is developed in Cao et al. [39] paper, utilizing the soil behavior type index profile computed from cone penetration test data. According to the results, the suggested method correctly determines the most likely soil stratigraphy based on the soil behavior type index profile and previous knowledge. The effect of structural health monitoring on a simulated sheet pile wall system's failure probability was examined by Chai et al. [40]. They estimated non-directly observable soil parameters from measured structural responses using Bayesian statistics. Their findings suggest that using Bayesian statistics in conjunction with structural health monitoring is a viable strategy to significantly lower our level of uncertainty when modeling hydraulic structures and, consequently, raise the calculated safety of those structures.

2.3 Foundation Bearing Capacity and Settlement

A new hybrid artificial neural network and mathematical model was proposed by Aouadj and Bouafia [41] to enhance the capacity to predict the load-settlement

behavior of shallow foundations in sandy soils. The proposed model was developed and validated using 110 full-scale loading tests of shallow foundations conducted in sand with cone penetration test (CPT) results. In terms of root mean square error, it was discovered that the suggested model outperforms traditional techniques by a margin of over 29%. Millán et al. [42] research adopted an ANN solution to predict the bearing capacity due to general shear failure obtained from numerical calculations based on the Hoek and Brown criterion. The predictions from the ANN model are in agreement with the numerical results.

Jebur et al. [43] looked at pile bearing capacity to create a solid model for simulating pile load-settlement behavior with a novel ANN technique. With a comparatively negligible mean square error level (MSE) of 0.0019, the proficiency metric indicators showed an exceptional degree of agreement between the measured and predicted pile-load settlement, resulting in a correlation coefficient (R) and root mean square error (RMSE) of 0.99, 0.043, respectively. ANN models can effectively predict the ultimate bearing capacity of a circular foundation on the sand layer of limited thickness when subjected to eccentric and inclined load [44]. Gnananandarao et al. [45] applied ANN and multivariable regression analysis (MRA) to predict the bearing capacity and the settlement of multi-edge skirted footings on sand, they found that the predicted bearing capacity ratio and settlement reduction factor for the multi-edge skirted footings with the use of ANN is superior to MRA.

The work of Pramanik et al. [46] uses the fuzzy set theory concept to analyze reliability based on foundation failure against bearing capacity. According to their findings, the soil's friction angle has the most significant impact out of all the variables. A fuzzy-based model hybridized with particle swarm optimization effectively estimated the bearing capacity of strip foundations rested on cohesionless slopes [47]. Similarly, the hybridized fuzzy model showed the highest predictability performance against all employed models in Mohammed et al.'s [48] study, which focused on quantifying shallow foundation settlement.

GA optimizes other soft computing models to solve bearing capacity and settlement problems. Here are some examples:

a. It optimizes the adaptive neuro-fuzzy inference system (ANFIS)–polynomial neural network (PNN) structure in Jahed Armaghani et al.'s [49] work to produce more accurate results than other methods considered for the load-carrying capacity assessment of thin-walled foundations.
b. In Liu et al.'s [50] work, GA-ANN models performed best in estimating the maximum settlement of eco-friendly raft-pile foundation systems amongst conventional ANN and fuzzy logic models.
c. ANN was optimized using GA and particle swarm optimization (PSO) to predict pile-bearing capacity. The results showed that both hybrid approaches could predict bearing capacity with a high degree of accuracy; however, the PSO-ANN predictive model is more useful in terms of performance capacity. This is due to a higher R^2 performance index [51].

In order to calibrate the resistance factor for the pile foundation design, Zhang et al. [52] study set out to create a Bayesian network-based machine-learning technique

that would generate site-specific statistics of the model bias factor using data from both the regional and site-specific load test data. It was discovered that a small number of site-specific pile load test results could significantly lower the uncertainty related to the model bias factor, increasing the pile design's cost-effectiveness.

Christodoulou and Pantelidis [53] focused on specific field research to achieve an ideal design, attempting to lower statistical uncertainty in the elastic settlement analysis of shallow foundations. Based on an advanced probabilistic framework that takes into account sampling of soil property values and applies the Random Finite Element Method (RFEM), their work is conducted. This work demonstrates that the appropriate sampling strategy can greatly decrease the statistical error.

3 Applications in Water Resources Engineering

3.1 Discharge Forecasting

The performance of artificial neural networks (ANN), which are being used increasingly for flood forecasting, depends on the choice of relevant inputs [54]. Hadiyan et al. [55] applied static and dynamic artificial neural networks for forecasting inflow discharges and found that their results provide useful information for reservoir inflow simulation. According to Dtissibe et al. [56], artificial neural networks schemes, as opposed to physical-based flood forecasting methods, consider all the parameters involved in the flood modeling, which can vary along a channel. ANN helps accurately model the rainfall-runoff process for catchments [57].

Artificial neural network (ANN) models were created and trained using various rainfall patterns in Ali and Shahbaz's [58] work to forecast daily river stream flow. The results showed that the ANN model created by presenting rainfall patterns of the previous four days can accurately predict the daily stream flow with values of R^2 for the validation and test periods of 0.97 and 0.94, respectively.

Fuzzy logic can be implemented in river basins where adequate hydrologic data is available, but it is not good enough for a more sophisticated model [59]. Tabbussum and Dar [60] proposed an efficient fuzzy inference system to improve floods' real-time forecasting. The reliability and robustness of the fuzzy logic approach were demonstrated with daily and 6-hourly discharge predictions in 4 rivers in 3 countries having contrasting climatological, geographical, and land use characteristics [61]. As illustrated in Fig. 3, a fuzzy logic system was among the machine learning frameworks considered for flood forecasting in Puttinaovarat and Horkaew [62] work.

Nguyen et al. [63] forecast discharge accurately using GA. They specifically used GA to suggest a novel mechanism that can automatically identify the ideal hyper-parameters for their prediction model. According to their findings, the suggested solution performs better than existing methods in various metrics. Young et al. [64] study used HEC-HMS (Hydrological engineering center hydrological modeling system)

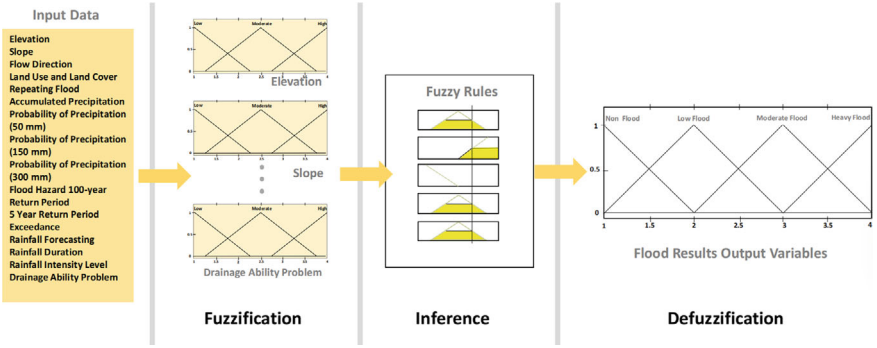


Fig. 3 Diagram of a Fuzzy Logic employed in flood forecasting [62]

model combined with GANN (Genetic algorithm neural network) and ANFIS (Adaptive neuro-fuzzy inference system) to predict the runoff discharge of a watershed. As depicted in Fig. 4, the results were confirmed with several statistical indicators and proved to predict accuracy better.

Chen and Yu [65] developed a method which involves a deterministic stage forecast generated from support vector regression and a probability distribution of forecast error based on the fuzzy inference model. The effectiveness of the suggested

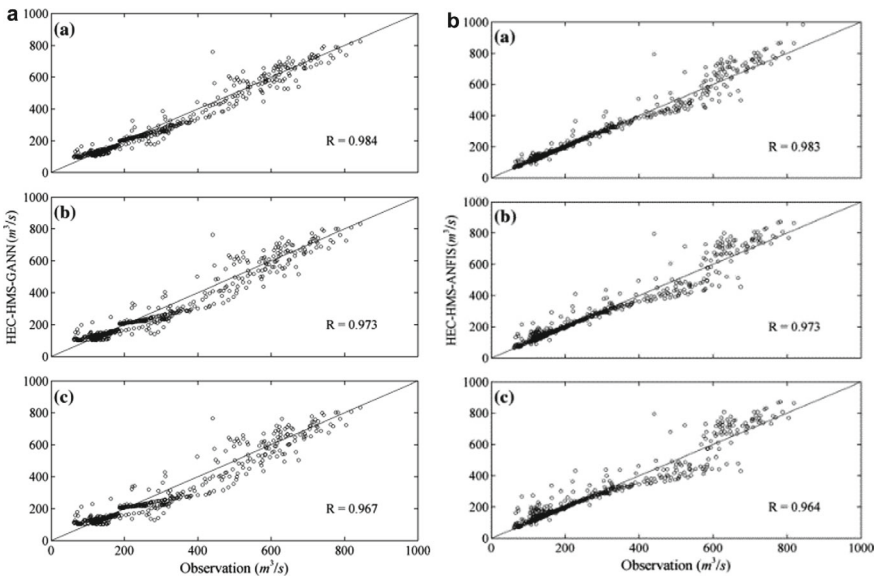


Fig. 4 **a** Scatter plots of predicted and measured runoff discharges for the HEC-HMS–GANN model validation: (**a**) 2-h, (**b**) 4-h, and (**c**) 6-h ahead predictions [64]. **b** Scatter plots of predicted and measured runoff discharges for the HEC-HMS–ANFIS model validation: (**a**) 2-h, (**b**) 4-h, and (**c**) 6-h ahead predictions [64]

methodology is demonstrated by the forecasting results analyzed by forecast hydrographs with a 95% confidence interval and the percentages of data contained in the confidence region. Compared to physically based models that used ensemble prediction approaches, the probabilistic modeling methodology of Garrote et al. [66], which used Bayesian networks to explain hydrologic processes, produced shorter computing times.

3.2 Water Quality Prediction

Advanced artificial intelligence (AI) algorithms such as nonlinear autoregressive neural network (NARNET), etc., in Aldhyani et al.'s [67] study have proved their capabilities to accurately predict water quality index (WQI) and classify water quality. The performance of machine learning techniques such as Random Forest (RF), ANN, Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), and Bagged Tree Model (BTM) to predict the water quality components of an Indian water quality dataset was evaluated in this work [68]. From the findings, the ANN model, as illustrated in Fig. 5, outperformed other models except the SVM).

Noori et al. [69] combined a process-based watershed model and ANN to improve the water quality predictions in unmonitored watersheds.

Fuzzy logic can be applied to accurately assess groundwater quality and eliminate inherent errors [70]. In contrast to prevalent indices like the National Sanitation Foundation Water Quality Index (NSFWQI) and the Vedprakash Water Quality Index (VWQI), Nayak et al. [71] found that the values of the fuzzy-based water quality index are more representative of the actual river water quality status of Indian rivers. Jha et al. [72] proposed a novel hybrid framework for evaluating groundwater quality and its spatial variability that combines Fuzzy Logic with the Geographical Information System (GIS) based Groundwater Quality Index (GQI). The fuzzy logic-based decision-making approach (FGQI) was found to be more practical and dependable when assessing and analyzing groundwater quality at larger scales, like basin or watershed scales. Mallik et al. [73] equally analyzed groundwater suitability for

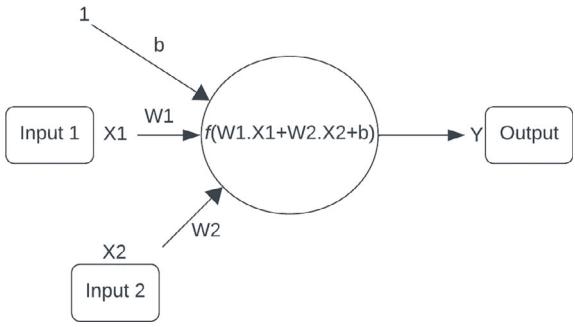


Fig. 5 Working procedure of the ANN model [68]

drinking by utilizing fuzzy logic in the geographical Information System (GIS) platform. Ghorbani et al. [74] set out to determine the value of the quality index as the objective function integrated into the fuzzy set theory so that it could reduce the uncertainties related to water quality goals and quickly specify the river's water quality status.

Using GA, Liu et al. [75] tried to calibrate a diffuse pollution model intended to simulate the export of phosphorus from point sources (humans) and diffuse sources (agricultural land). Yang et al. [76] used a genetic algorithm in conjunction with a Bayesian method to enhance sampling performance in the parameter calibration stage of their study, which assessed water quality.

In order to assess and interpret a sizable, complex matrix of water quality (WQ) data gathered in the Paute river basin, GA-based algorithms were used [77]. Swain and Sahoo [78] used satellite data products and a GA processing technique to enhance river water quality monitoring. The main goal of Habiyakare et al. [79] research was to ascertain the source and extent of pollution of migrating dense non-aqueous phase liquids (DNAPLs) by employing a combination of GA and numerical simulation.

Peng et al. [80] proposed a Bayesian Joint Probability (BJP) modeling approach that addresses bias correction and total uncertainty quantification for daily forecasts of water quality parameters derived from dynamical lake models. Panidhapu et al. [81] explored a novel application of Bayesian Belief Networks (BBNs) for real-time modeling of surface waters containing fecal indicator bacteria (FIB) levels. BBNs allow for a probabilistic representation of complex variable interactions. Advantageously, predictions from incomplete monitoring data and probabilistic inference of variable importance in fecal indicator bacteria (FIB) levels were found to be produced by the BBN approach. When Farjoudi et al. [82] used the probabilistic bankruptcy method, as shown in Fig. 6, to manage river water quality, they concluded that the model was reliable and could be used in situations where parties did not cooperate, providing greater flexibility in real-world scenarios.

Zhou [83] used a multivariate Bayesian uncertainty processor (MBUP) to model the relationship between the point forecasts from a deep-learning artificial neural network (ANN) and the corresponding observed water quality in a probabilistic manner. It is interesting to note that even in cases where the missing rate of input data reaches 50%, the approach managed to extract the intricate dependence structure between the model's output and the observed water quality. Yu and Zhang [84] study proposed a copula-based Bayesian network (CBN) method to assess the water quality risk in a large drinking water reservoir in a tangible manner using multiple environmental risk indicators.

3.3 Water Treatment Process

Taloba [85] developed an ANN to forecast the performance of a reverse osmosis desalination process, which was subsequently applied to water temperature modeling. To simulate the performance of the Tamburawa water treatment plant

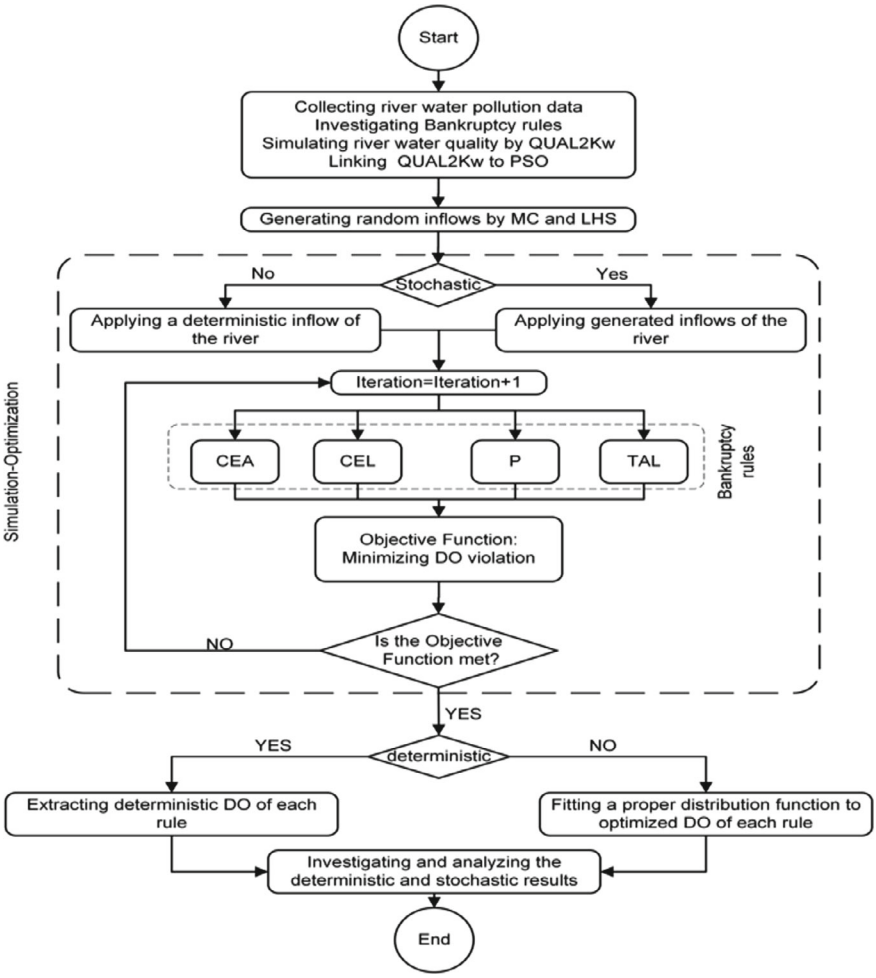


Fig. 6 Flowchart of the proposed probabilistic bankruptcy method [82]

in terms of pH and turbidity, Abba et al. [86] used ANN. The overall results of the models showed the reliability and satisfactory performance of ANN in modeling the performance efficiency of the water treatment plant (WTP). Solaimany-Aminabad et al. [87] predicted the influent water quality of the Sanandaj water treatment plant (WTP) using an ANN approach and a feed-forward back-propagation non-linear autoregressive neural network. The predicting ANN model offers an efficient diagnostic and analysis tool to comprehend and replicate the non-linear behavior of the influent water characteristics. Tashaouie et al.'s [88] also utilized ANN in predicting the performance of pressure filters in a water treatment.

Santín et al. [89] presented a novel control approach that utilizes a fuzzy controller to adjust the internal recirculation flow rate in wastewater treatment plants. The results

demonstrate improvements in reducing ammonia limit violations, total nitrogen limit violations, and pumping energy costs by implementing the suggested fuzzy controller. The uncertainty and challenges that arise when choosing an effective treatment strategy are mitigated by linear diophantine fuzzy sets [90]. The consistency and resilience of decision-making in these drinking WTP can be enhanced using fuzzy inference systems, which can aid in consolidating the process of knowledge gained through years of experience [91]. An adaptive neuro-fuzzy inference system (ANFIS) is suggested by Okoji et al. [92] to forecast trihalomethane (THM) levels in real distribution systems. ANFIS demonstrated efficacy in anticipating THM formation based on the statistical indices obtained, thereby facilitating enhanced monitoring of disinfection by-products (DBPs) in water treatment systems.

Findings from the work of Gupta and Shrivastava [93] show that using GA in conjunction with Monte Carlo simulation (MCS) is a reliable way to consistently arrive at the optimal or nearly optimal solution for the reliability constrained WTP design problem. In Swan et al. [94] study, operational regimes for water treatment were optimized using verified static and dynamic models of operational works in conjunction with GA and Monte Carlo conditions. In Al-Obaidi et al.'s [95] work, GA was developed for a reverse osmosis wastewater system performance to solve a formulated optimization problem involving two objective functions, which are to maximize the solute rejection at different cases of feed concentration and minimize the operating pressure to improve economic aspects. An intelligent system with GA-based operations with fitness values and a neural network for training was proposed by Sophia et al. [96]. By calculating the objective function of different population types, the system can be used to increase optimization performance by making predictions about water consumption and distribution using decision-making algorithms.

The study by Zhu and McBean [97] showed that Bayesian decision networks (BDNs) could offer a normative framework to help with WTP problem solving. Bertone et al. [98] created a data-driven Bayesian Network (BN) model for a sizable drinking water treatment facility in Australia to precisely forecast the likelihood of various incoming raw water quality ranges and evaluate various scenarios (such as flow and timing) of dam water releases. When evaluating various WTP control strategies in Flores-Alsina et al. [99] study, the activated sludge uncertainty introduced was characterized by probability distribution functions using available process information.

4 Applications in Transportation Engineering

4.1 Transportation Infrastructure

According to Marovic et al. [100], ANNs can be trained to accurately predict road deterioration, enabling them to be utilised as a tool for maintenance planning activities for transportation infrastructure. It was found that applying such a model in

the decision support concept is possible and desirable to improve urban road infrastructure maintenance planning processes. A decision support concept (DSC) aimed at improving urban road infrastructure planning based on multicriteria methods and artificial neural networks proposed by Jajac et al. [101] showed how urban road infrastructure planning can be improved. It showed how decision-making processes during the planning stages can be supported at all decision-making levels by proper interaction between DSC modules.

Abu lebdeh et al. [102] presented an overview of different techniques to improve the performance of GAs, with particular emphasis on parallel GAs (PGAs). They stated that using parallel GAs does not reduce the importance of seeking efficient problem-specific operators and parameter values. However, it magnifies such choices' effectiveness and increases the range of options available. The advantages PGAs offer mean more efficient and faster optimization for many applications in civil infrastructure design, operating management, and maintenance projects. Efthymiou et al. [103] presented a GA approach. Since data from electric vehicles usage are still scarce, origin—destination data of conventional vehicles are used, and the necessary assumptions to predict electric vehicles' penetration in the years to come are made. The algorithm and a user-friendly tool have been developed in R and tested for the city of Thessaloniki. The results indicated that 15 stations would be required to cover 80% of the estimated electric vehicles charging demand in 2020 in the city of Thessaloniki, and their optimal locations to install them are identified.

Inti et al. [104] proposed modifying AHP using the additive transitivity property of fuzzy preference relations because it significantly reduces the number of inputs and minimizes inconsistency. The effectiveness of the proposed method was verified through a case study for selecting a contractor from six contractors. Inputs were taken from three decision makers for both traditional and proposed AHP, and the rankings obtained by the proposed method were compared with traditional AHP. The comparison revealed that using fewer inputs in conjunction with additive transitive fuzzy preference relations generated consistent judgments in minimal time. Abeysekara et al. [105] explored the factors that pose risks to the efficiency with which capital is utilized in large-scale transport infrastructure investments and suggests risk management best practices that may be applied in relation to each. Fuzzy set theory was applied to analyze and evaluate risks and the effect of adopting best practices in relation to them. By linking each risk to a best practice, potential enhancements to capital usage efficiency in infrastructure investment projects are observed. They concluded that risks in large scale transport infrastructure investments may be managed through efficiency-enhancing practices to reduce.

The vast majority of the cost overrun literature has tended to adopt a deterministic approach in examining the phenomenon's occurrence; this paper proposes a shift towards adopting a pluralistic probabilistic approach to cost overrun causation [106]. A study by Van Noortwijk and Frangopol [107] described and compared two probabilistic life-cycle maintenance models that are currently being used to balance structural reliability optimally and the life-cycle cost of deteriorating civil infrastructure. Frangopol's model contributed to further developing the bridge maintenance

methodology that the UK Highways Agency set up, whereas Rijkswaterstaat's model has been applied in the Netherlands.

4.2 Traffic Flow Modelling and Control

Kranti Kumar et al. [108] applied Artificial Neural Network for short-term prediction of traffic volume using past traffic data. Besides traffic volume, speed, and density, the model incorporates both time and the day of the week as input variables. The model has been validated using actual rural highway traffic flow data collected through field studies. Artificial Neural Network have produced good results in this study even though the speeds of each category of vehicles were considered separately as input variables.

Teklu et al. [109] considered the problem of optimizing signal green and cycle timings over an urban network, in such a way that the optimization anticipates the impact on traffic routing patterns. GA was devised to solve the resulting problem, using total travel time across the network as an illustrative fitness function and with a widely used traffic simulation-assignment model providing the equilibrium flows. The results show a better performance of the signal timing as optimized by the GA method as compared to a method that does not consider rerouting. Mesbah et al. [110] aimed to reallocate the road space between private cars and transit modes so that the system is optimized. A GA approach is used, which enables the method to be applied to large networks. The application of a parallel GA is also demonstrated in the solution method, which has a considerably shorter execution time. It was found that the proposed methodology can successfully consider the benefits to all stakeholders in introducing transit lanes.

In this work by Peter et al. [111], an attempt to improve upon an existing programmed stationary road traffic light control system of the Kaduna Refinery Junction (KRJ) is considered. An efficient fuzzy logic (FL) model is developed for the optimal traffic light control system scheduling using TraCI4MATLAB and Simulation of Urban Mobility (SUMO). An average improvement of 2.74% over an earlier result was obtained. Considering priority for emergency vehicles, an improvement of 66.79% over the static phase scheduling was recorded. Jabari and Liu [112] presented a new stochastic model, for which a long-run temporal mean dynamic is shown to be consistent with a well-established first-order traffic flow model, the CTM. At the same time, the proposed model implicitly ensures the non-negativity of traffic densities.

4.3 Transport System Network Design Interface

Huang et al. [113] presented a framework of bus network optimization based on Geographical Information Systems (GIS) and GA. The GA process for bus network

optimization has been improved by incorporating trip demand at bus stops. Trip generation and trip attraction at bus stops are utilized in Candidate Routing (CR) creation and GA optimization. The bus route network created from the framework provides transit planners and policy makers with a preliminary solution set, which can be utilized as a starting basis for route deployment. Ernesto et al. [114] proposed a procedure for solving the bus network design problem in a large urban area characterized by a multimodal transit system. The solving procedure consists of a set of heuristics, including a first route generation routine based on the flow concentration process and a parallel genetic algorithm for finding an optimal or near-optimal network of routes with the associated frequencies.

An-hu et al. [115] applied a fuzzy neural network algorithm to the intersection traffic management system, aiming at alleviating traffic congestion and improving the efficiency of intersection traffic. The fuzzy neural network system is based on fuzzy logic theory, combined with neural network self-correction method and self-organization, to achieve the purpose of information processing. Through delay comparison, it was found that the average delay of the fuzzy control relative timing control data shows that the average delay of the low peak vehicle is reduced by 8.8%, the mid-peak period is reduced by 11.2%, and the peak period is decreased by 11.8%; while the fuzzy neural network is relatively fuzzy. Koukol et al. [116] aimed to provide a comprehensive overview of the literature on fuzzy control systems used to manage the road traffic flow at road junctions. They conducted a comprehensive review of the literature dealing with the use of fuzzy sets and fuzzy logic theory in the field of traffic control systems. The review focused on various approaches that describe and predict the driver's behaviour and optimize the flow of traffic.

4.4 Public Transport Management

Deng et al. [117] presented a revised Importance–Performance Analysis (IPA) that integrates a back-propagation neural network and three-factor theory to effectively assist practitioners in determining critical service attributes. Finally, a customer satisfaction improvement case is presented to demonstrate the implementation of the proposed Back-Propagation Neural Network based Importance–Performance Analysis (BPNN-IPA) approach.

The method used in this study [118] is sentiment analysis, which uses genetic algorithms for feature selection with comparative classification algorithms. It was found that the Support Vector Machine classification algorithm based on Genetic Algorithms had a reasonably good average accuracy of 76.11% and an AUC value of 0.778% with a Fair Classification diagnosis level compared to the three methods such as Naive Bayes, Support Vector Machine and Naive Bayes based on Genetic Algorithms. Serban et al. [119] analyzed the deviations from the planned (declared and publicly) schedule for a tram line in Bucharest city at different stations, at different times of the day, and at different days of the week in order to build an optimisation model for needed adjustments to the planned schedule. The data are collected using

the Automatic Vehicle Location (AVL) system installed on the trams' board. The number of adjustment solutions is very large, and the genetic algorithm is engaged for the optimisation model solving.

This work by Naumov et al. [120] contributed to the direction of using fuzzy logic to assess travelers' preferences. The aim was to develop a simple but reliable method that uses the travelers' survey data to calculate the membership functions describing the basic preferences of the passengers of a public transport system: pricing, comfortability, and travel speed. After tuning the controller parameters and conducting several simulation tests, they obtained promising results in terms of savings in waiting times with the implementation of the proposed rules, noting that the best performance occurred when fuzzy rules were included.

4.5 Autonomous Vehicle and Advanced Driver Assistance Systems

Klück et al. [121] studied using a genetic algorithm to test parameter optimization in the context of autonomous and automated driving. Their approach iteratively optimizes test parameters to obtain critical scenarios that form the basis for virtual verification and validation of Advanced Driver Assistant Systems (ADAS). Their genetic algorithm approach showed a higher chance of generating a critical scenario than a random selection of test parameters.

Al-Saad et al. [122] presented an intelligent driver assistance system, including adaptive cruise control (ACC) and an energy management system (EMS), for HEVs. Their proposed ACC determines the desired acceleration and safe distance with the lead car through a switched model predictive control (MPC) and a neuro-fuzzy (NF) system. The results showed that the driving risk is extremely reduced using ACC-MPC and ACC-NF, and the vehicle energy consumption by driver assistance system based on ACC-NF is improved by 2.6%. Ghahroudi et al. [123] proposed a network of sensors for advance driver assistance systems with a specific deployment on a host vehicle integrated with a logical, effective, and practical hybrid sensor fusion technique using a fuzzy method which is applicable in various depth of fusion for high-speed vehicles in roads and highways. The control section, the fuzzy logic system (FLS), played an acceptable rule, much better than an individual driver, to control the vehicle safely and be observant in encountering unexpected obstacles. The results improved with an exponentially moving average window filter.

5 Applications in Environmental Engineering and Public Health

5.1 Air Quality Modeling

In Alimissis et al. [124] study, the spatial variability of air pollution is modeled using the MLR and ANN approaches. The evaluation of both linear (MLR) and non-linear (ANN) schemes is performed using the leave-one-out cross-validation methodology, where for each pollutant, a specific monitoring site is the target site and the concentrations at the remaining monitoring sites (their number depends on the pollutant) are used to estimate the air pollutant concentrations at the target site. The results highlight the superior performance of the FFNN models compared to the linear MLR interpolation scheme due to their ability to model more efficiently complex air.

Relvas and Miranda [125] described the design and application of a modeling system capable of rapidly supporting decision-makers regarding urban air quality strategies, in particular, providing emission and concentration maps, as well as external costs (mortality and morbidity) due to air pollution, and total implementation costs of improvement measures. Results from a chemical transport model were used to train artificial neural networks and link the emission of pollutant precursors and urban air quality. The Integrated Urban Air Pollution Assessment Model (IUAPAM) was applied to Porto city (Portugal), and results showed that it is possible to reduce the number of premature deaths per year attributable to particulate matter (PM10) from 1300 to 1240 (5%), with an investment of 0.64 M€/year, based on fireplace replacements.

Li et al. [126] proposed the stepwise genetic algorithm (SGA), which is adapted from GA to satisfy the optimization requirements under a framework of adaptive management. SGA is applied to optimize the Air quality monitoring network (AQMN) of mainland China by identifying an array of sites for removal consideration, which is presented in ascending order based on the impact of each site on the MKV of annual average PM2.5 concentrations (Fig. 7).

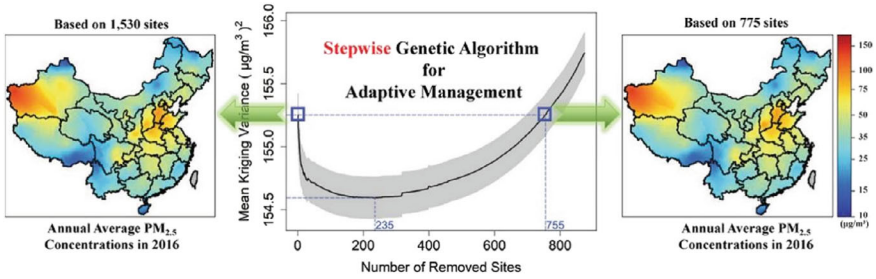


Fig. 7 Stepwise genetic algorithm [126]

Ma and Zhang [127] presented a model that can aid planners in defining the total allowable pollutant discharge in the planning region, accounting for the dynamic and stochastic character of meteorological conditions. This is accomplished by integrating Monte Carlo simulation and using a genetic algorithm to solve the model. To evaluate the model's effectiveness, the approach's results are compared with those of the linear deterministic procedures. They also provided valuable insights about how air quality targets should be made when the air pollutant will not threat the residents' health.

This study by Javid et al. [128] aimed to develop a novel, fuzzy-based index (FIAQ) for assessing the air quality in indoor environments. For this purpose, we considered three crucial categories of indoor air pollutants, namely, criteria air pollutants, volatile organic compounds, and bioaerosols, in the body of the index. In addition, a case study of virtually generated indoor environments was also provided to indicate the index performance. According to the results from the above study, the FIAQI can be considered a more useful, comprehensive tool to classify the IAQ compared to the current methods of IAQ assessment, which rely mainly on the evaluators' observations or quantitative measurement of a single quality parameter of the IAQ (Fig. 8).

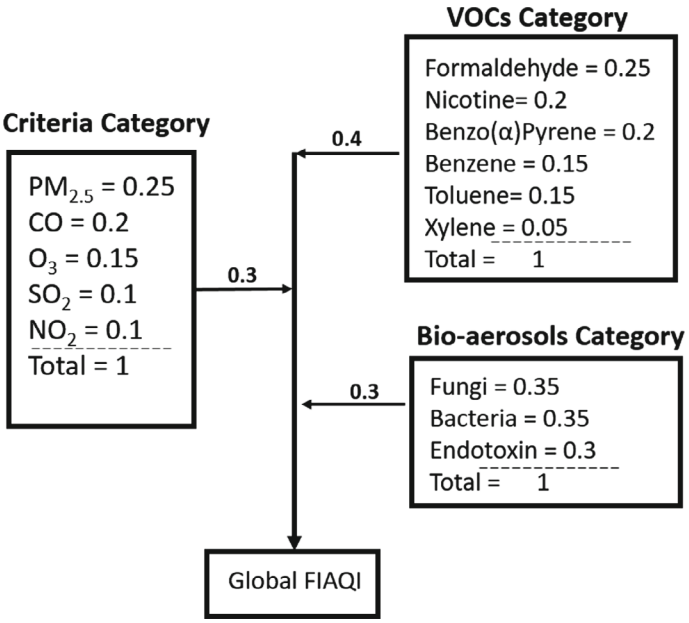


Fig. 8 Weighting assignment to different parameters and groups included in the FIAQI [128]

5.2 Health Risk Assessment

Sasaki et al. [129] predicted EMS cases for 5-year intervals from 2020 to 2050 by correlating current EMS cases with demographic factors at the census area level and predicted population changes. It then applies a modified grouping genetic algorithm to compare current and future optimal locations and numbers of ambulances. Future EMS demands were predicted to increase by 2030 using the model ($R2 = 0.71$). The optimal locations of ambulances based on future EMS cases were compared with current locations and with optimal locations modelled on current EMS case data. Optimising the location of ambulance stations locations reduced the average response times by 57 seconds.

Gül et al. [130] conducted at a prominent hospital in Turkey, the authors aimed to assess occupational health and safety (OHS) risks for health staff. They employed a two-stage fuzzy multi-criteria approach, combining the Fuzzy Analytic Hierarchy Process (FAHP) and the fuzzy VIKOR (FVIKOR) method. The FAHP was used to weigh five risk parameters: severity, occurrence, undetectability, sensitivity to maintenance, non-execution, and sensitivity to personal protective equipment (PPE) non-utilization. Subsequently, the FVIKOR approach prioritized hazard types in each department of the hospital. The study concluded by implementing hazard control measures and identifying areas for improvement.

5.3 Disease Spread Prediction

Wang et al. [131] developed a method to spectrally predict late blight infections in tomatoes based on an artificial neural network (ANN). The ANN was designed as a back-propagation (BP) neural network that used a gradient-descent learning algorithm. Results of discrete data indicated different levels of disease infestations. The correlation coefficients of prediction values and observed data were 0.99 and 0.82 for field data and remote sensing image data, respectively. Laureano-Rosario et al. [132] applied artificial neural networks (ANNs) to predict dengue fever outbreak occurrences in San Juan, Puerto Rico (USA), and in several coastal municipalities of the state of Yucatan, Mexico, based on specific thresholds. The models were trained with 19 years of dengue fever data for Puerto Rico and six years for Mexico. The predictive power was above 70% for all four model runs. The ANNs were able to successfully model dengue fever outbreak occurrences in both study areas.

Traulsen and Krieter [133] evaluated fuzzy logic as a modelling technique for predicting the airborne spread of foot and mouth disease. Using linguistic variables, livestock-specific fuzzy logic models were developed. A good agreement was found between Gaussian dispersion and fuzzy logic models, with no limitations to certain livestock. Lefevr et al. [134] looked at the problem of HIV spreading in needle drug individuals, analyzing the dynamic procedures generated in corresponding complex networks. The authors propose an epidemic model based on fuzzy logic, aiming to

establish a relation between viral load and clinical evolution to Acquired Immunodeficiency Syndrome (AIDS) in HIV-contaminated users. The model is based on the Erdös-Renyi model and can be applied to many different spreading processes in complex networks. The study uses four different scenarios, including the number of users and the number of tests per year. The results show that the first scenario is more Fuzzy than the second scenario, with the first scenario being more Fuzzy. The study also explores the impact of contact structure on the transmission of HIV (Fig. 9).

Monteiro et al. [135] proposed an epidemic model based on probabilistic cellular automaton to predict the prevalence of varicella in Belgium and Italy around 2000. The model uses a genetic algorithm to identify three parameters based on data from the pre-vaccination period. The results showed that the model can predict varicella prevalence with an average relative error of 2–4%. Johnson et al. [136] showed that early diagnosis and risk assessment of Alzheimer's disease (AD) were crucial for prevention. Genetic algorithms (GA) were used to search for a combination of variables efficiently, enhancing diagnosis accuracy. GA's performance in predicting disease progression was superior to single significant variables.

Zervoudakis et al. [137] presented an approach that exploits information available on social media to predict whether a patient has been infected with COVID-19. Our approach is based on a Bayesian model trained using data collected online. Then, the trained model can be used to evaluate the possibility that new patients will be infected with COVID-19. The experimental evaluation presented shows the high quality of our approach. In addition, our model can be incrementally retrained to become more robust and efficient.

5.4 Optimal Resource Allocation

Phan et al. [138] reviewed the applications of BBNs to water resource management, identifying gaps in their application and examining methods for developing and validating BBN models. A systematic quantitative literature review was conducted to identify and evaluate studies on Bayesian Belief Networks (BBNs) for water resource management.

Liao et al. [139] proposed a multi-scale land use optimization method based on benefit coupling evaluation, BP-ANN, and CLUE-S models for China's rural production-living-ecological (PLE) space. The study found a high correlation between land use patterns and benefits, reducing production land by 8.94% and increasing ecological land by 9.2%. The method supports the sustainable use of rural land resources.

Mohaddes et al. [140] proposed mathematical programming of a fuzzy land-use optimization model with economic, social, and environmental objectives and implications in regions. They confirmed that the present fuzzy multi-objective model is more appropriate than non-fuzzy problem formulation in reflecting a realistic situation (Fig. 10).

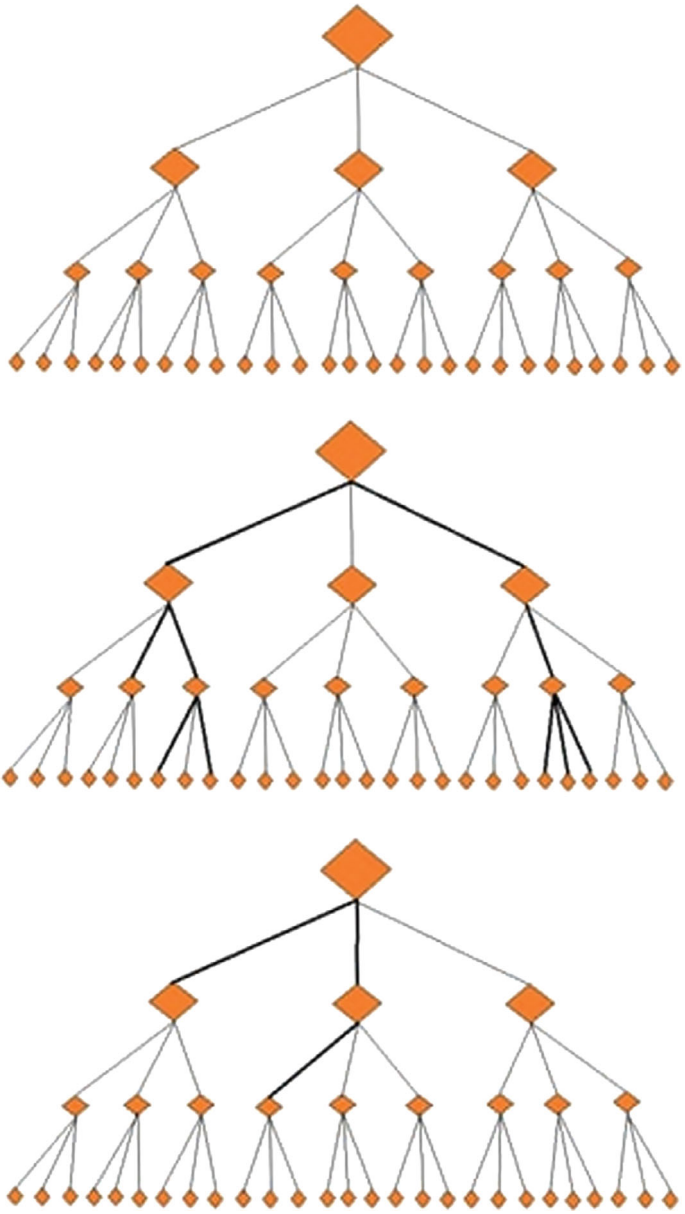


Fig. 9 A tree network of an epidemic spread [134]

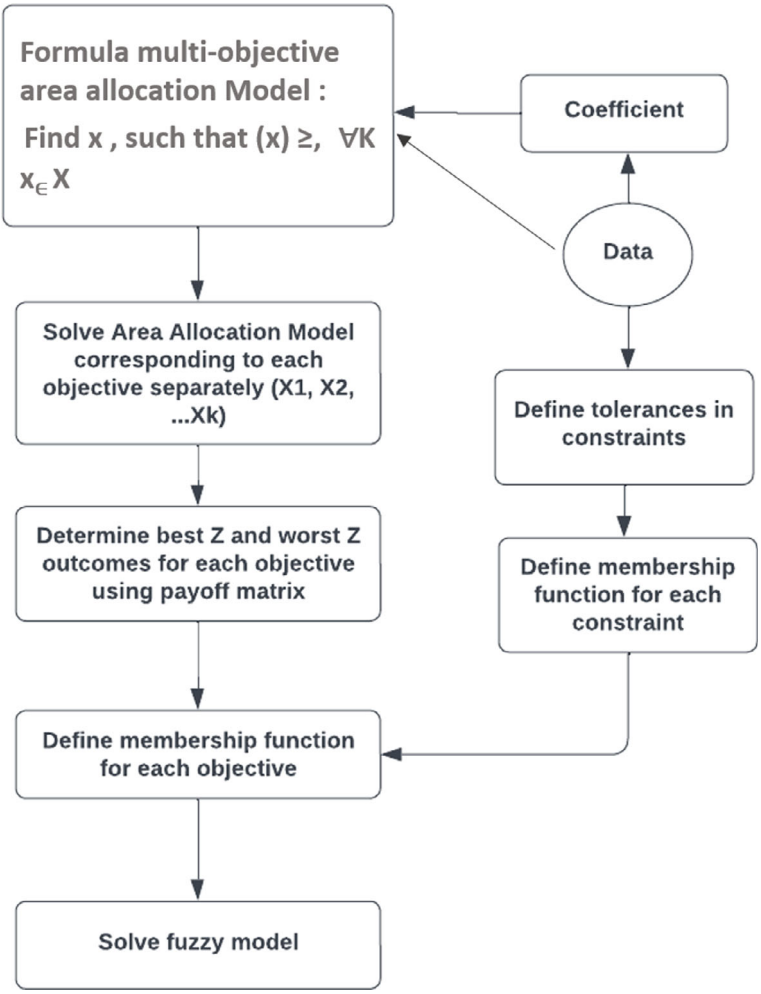


Fig. 10 Flowchart of fuzzy approach methodology [140]

6 Applications in Structural and Construction Engineering

Cement-based composites are the most commonly utilized materials for building and construction. The composition of this composite material is usually a function of the purpose of construction and availability of raw materials. Researchers have made continuous attempts to develop high-performance and multifunctional cement composite materials. However, these attempts face setbacks since the traditional investigation approach requires intensive labor and time. In addition, the application of already existing primitive models often makes it difficult to understand the behavior of these materials.

Huang et al. [141] adopted the machine learning approach in predicting the mechanical properties of carbon nanotube-reinforced cement composites. Machine learning is a subset of AI that accurately predicts desired outputs. It is a computer-based approach that recognizes and learns patterns from existing data with little or no supervision. Awolusi et al. [142] also applied the machine learning approach by artificial neural networks to predict steel fiber-reinforced concrete. Five different algorithms which are Genetic algorithm, Levenberg Marquardt, quick propagation, incremental back propagation, and batch back propagation. The general observations of the aforementioned studies provide more efficient and accurate means for predicting the properties of cement composites when compared to the regression models and traditional experimental approach. Another application of the machine learning approach, also referred to as artificial intelligence, is the application of fuzzy logic in building.

Taneva et al. [143] applied fuzzy logic to automated temperature control of buildings. The fuzzy control is based on the fuzzy linguistic variables that utilize the concept of fuzzy logic. A model was achieved for the room heating system, and simulated results were utilized in the actual building management strategy. Azizi et al. [144] applied a crystal structure algorithm for optimizing vibration control in structural engineering. The approach is based on the structural design principles of crystalline solids containing lattice as the primary component. The lattice represents a periodic array of imaginary points in a predefined space. It utilizes the optimization of fuzzy logic controllers in buildings to demonstrate its ability to solve real life engineering problems. The two real life structures were considered. The first was a three stories building, and the other a twenty stories building, seismically induced vibrations were intelligently controlled using the fuzzy logic controllers. It was concluded that the crystal structure algorithm approach was effective in the vibration control of buildings. Zohrabzadeh et al. [145] applied fuzzy logic in controlling vibration in buildings with the aid of fuzzy and neuro-fuzzy systems. The neuro-fuzzy inference system was able to regulate function parameters, while the fuzzy controller was able to reduce the damages in buildings by 20%.

Tosee et al. [146] predicted the crack width in reinforced concrete as a one way slab strengthened with carbon fiber reinforced polymer using the hybridized grey wolf optimizer neural network model. This approach utilized both ANN and Grey Wolf Optimization algorithms. In order to adequately predict the crack, the applied load, width-to-length ratio of the carbon fiber reinforced polymer, stress in steel reinforcement, and crack positions were considered as input parameters for the study. Another machine learning algorithm that is applicable in civil engineering is Gene Expression Programming (GEP). Tung et al. [147] applied the GEP to predict the residual compressive strength of concrete containing fly ash. The study achieved an efficient and accurate model which as applicable for predicting concrete exposed to elevated temperature. The prediction process of GEP exhibits transparency and insight by providing an empirical relationship between independent and dependent variables. For this reason, it is regarded as a grey box model compared to other machine learning algorithms. The GEP is an evolutionary algorithm that explores the best solution for a problem within a design space. It does not require optimization of the design spaces

as applicable to ANN and other machine learning algorithms. Therefore, GEP has the advantage of reducing computation time and labour. Furthermore, Tung et al. [148] also applied GEP in evaluating the post-fire mechanical properties of recycled aggregate containing ground granulated blast furnace slag. The GEP was able to optimize and predict all properties investigated with high accuracy.

References

1. Meng, J., Mattsson, H., Laue, J.: Three-dimensional slope stability predictions using artificial neural networks. Int. J. Numer. Anal. Methods Geomech. **45**(13), 1988–2000 (2021). https://doi.org/10.1002/nag.3252
2. Kostić, S., Vasović, N., Todorović, K., Samčović, A.: Application of artificial neural networks for slope stability analysis in geotechnical practice. In: 2016 13th Symposium on Neural Networks and Applications (NEUREL), pp. 1–6 (2016). https://doi.org/10.1109/NEUREL.2016.7800125
3. Qian, Z.G., Li, A.J., Chen, W.C., Lyamin, A.V., Jiang, J.C.: An artificial neural network approach to inhomogeneous soil slope stability predictions based on limit analysis methods. Soils Found. **59**(2), 556–569 (2019). https://doi.org/10.1016/j.sandf.2018.10.008
4. Chakraborty, A., Goswami, D.: Prediction of slope stability using multiple linear regression (MLR) and artificial neural network (ANN). Arab. J. Geosci. **10**(17), 385 (2017). https://doi.org/10.1007/s12517-017-3167-x
5. Sakellariou, M.G., Ferentinou, M.D.: A study of slope stability prediction using neural networks. Geo-techn. Geol. Eng. **23**(4), 419–445 (2005). https://doi.org/10.1007/s10706-004-8680-5
6. Marrapu, B.M., Kukunuri, A., Jakka, R.S.: Improvement in prediction of slope stability & relative importance factors using ANN. Geo-tech. Geol. Eng. **39**(8), 5879–5894 (2021). https://doi.org/10.1007/s10706-021-01872-2
7. Abdalla, J.A., Attom, M.F., Hawileh, R.: Prediction of minimum factor of safety against slope failure in clayey soils using artificial neural network. Environ. Earth Sci. **73**(9), 5463–5477 (2015). https://doi.org/10.1007/s12665-014-3800-x
8. Li, A.J., Khoo, S., Lyamin, A.V., Wang, Y.: Rock slope stability analyses using extreme learning neural network and terminal steepest descent algorithm. Autom. Construct. **65**, 42–50 (2016). https://doi.org/10.1016/j.autcon.2016.02.004
9. Kumar, A., Chauhan, V.B.: Advanced finite element limit analysis and machine learning for assessing the stability of square tunnels in rock slope. Transp. Infrastruct. Geotechnol. (2023). https://doi.org/10.1007/s40515-023-00338-7
10. Lim, K., Cassidy, M.J., Li, A.J., Lyamin, A.V.: Mean parametric monte Carlo study of fill slopes. Int. J. Geomech. **17**(4), 04016105 (2017). https://doi.org/10.1061/(ASCE)GM.1943-5622.0000812
11. Gao, W., Raftari, M., Rashid, A.S.A., Mu'azu, M.A., Jusoh, W.A.W.: A predictive model based on an optimized ANN combined with ICA for predicting the stability of slopes. Eng. Comput. **36**(1), 325–344 (2020). https://doi.org/10.1007/s00366-019-00702-7
12. Khan, N., Vijaya, R.B., Reddy, A.M., Kumar, M.P.: Reliability and fuzzy logic concepts as applied to slope stability analysis–a review. Int. J. Eng. Res. Appl. **5**(6 Part-2), 01–03 (2015)
13. Habibagahi, G., Meidani, M.: Reliability of slope stability analysis evaluated using fuzzy set approach. In: 5th International Conference on Civil Engineering, Ferdowsi University, Mashhad, Iran (2000)
14. Daftaribesheli, A., Ataei, M., Sereshki, F.: Assessment of rock slope stability using the Fuzzy Slope Mass Rating (FSMR) system. Appl. Soft Comput. **11**(8), 4465–4473 (2011)

15. Mohamed, T., Kasa, A., Taha, M.R., et al.: Fuzzy logic system for slope stability prediction. Int. J. Adv. Sci. Eng. Inform. Technol. **2**(2), 38 (2012)
16. Azarafza, M., Asghari-Kaljahi, E., Ghazifard, A., Akgün, H.: Application of fuzzy expert decision-making system for rock slope block-toppling modeling and assessment: a case study. Model. Earth Syst. Environ. **7**(1), 159–168 (2021). https://doi.org/10.1007/s40808-020-00877-9
17. Palazzolo, N., Peres, D.J., Bordoni, M., Meisina, C., Creaco, E., Cancelliere, A.: Improving spatial landslide prediction with 3D slope stability analysis and genetic algorithm optimization: application to the Oltrepò Pavese. Water **13**(6), **6** (2021). https://doi.org/10.3390/w13060801
18. Wang, H., Moayedi, H., Kok Foong, L.: Genetic algorithm hybridized with multilayer perceptron to have an economical slope stability design. Eng. Comput. **37**(4), 3067–3078 (2021). https://doi.org/10.1007/s00366-020-00957-5
19. Bui, X.-N., Nguyen, H., Choi, Y., Nguyen-Thoi, T., Zhou, J., Dou, J.: Prediction of slope failure in open-pit mines using a novel hybrid artificial intelligence model based on decision tree and evolution algorithm. Sci. Rep. **10**(1), 9939 (2020)
20. Yuan, C., Moayedi, H.: The performance of six neural-evolutionary classification techniques combined with multi-layer perception in two-layered cohesive slope stability analysis and failure recognition. Eng. Comput. **36**, 1705–1714 (2020)
21. Zolfaghari, A.R., Heath, A.C., McCombie, P.F.: Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis. Comput. Geotech. **32**(3), 139–152 (2005). https://doi.org/10.1016/j.compgeo.2005.02.001
22. Chen, F., Zhang, R., Wang, Y., Liu, H., Böhlke, T., Zhang, W.: Probabilistic stability analyses of slope reinforced with piles in spatially variable soils. Int. J. Approx. Reason. **122**, 66–79 (2020)
23. Ji, J., Zhang, W., Zhang, F., Gao, Y., Lü, Q.: Reliability analysis on permanent displacement of earth slopes using the simplified Bishop method. Comput. Geotech. **117**, 103286 (2020). https://doi.org/10.1016/j.compgeo.2019.103286
24. Chen, Y., Zhang, L., Liao, C., Jiang, M., Peng, M.: A two-stage probabilistic approach for the risk assessment of submarine landslides induced by gas hydrate exploitation. Appl. Ocean Res. **99**, 102158 (2020). https://doi.org/10.1016/j.apor.2020.102158
25. Zhang, W., Meng, F., Chen, F., Liu, H.: Effects of spatial variability of weak layer and seismic randomness on rock slope stability and reliability analysis. Soil Dyn. Earthquake Eng. **146**, 106735 (2021). https://doi.org/10.1016/j.soildyn.2021.106735
26. Rani, C.S.: Artificial Neural Networks (ANNS) for prediction of engineering properties of soils. Artif. Neural Netw. **3**(1) (2013)
27. Kiran, S., Lal, B.: Modelling of soil shear strength using neural network approach. Electron. J. Geotech. Eng **21**(10), 3751–3771 (2016)
28. Rashidian, V., Hassanlourad, M.: Application of an artificial neural network for modeling the mechanical behavior of carbonate soils. Int. J. Geomech. **14**(1), 142–150 (2014). https://doi.org/10.1061/(ASCE)GM.1943-5622.0000299
29. Zhang, N., Shen, S.-L., Zhou, A., Jin, Y.-F.: Application of LSTM approach for modelling stress–strain behaviour of soil. Appl. Soft Comput. **100**, 106959 (2021). https://doi.org/10.1016/j.asoc.2020.106959
30. Dewidar, A.Z., Al-Ghobari, H., Alataway, A.: Developing a fuzzy logic model for predicting soil inltration rate based on soil texture properties. Water SA **45**(3), 400–410 (2019)
31. Sujatha, A., Govindaraju, L., Shivakumar, N., Devaraj, V.: Fuzzy knowledge based system for suitability of soils in airfield applications. Civil Eng. J. **7**(1), Article 1 (2021). https://doi.org/10.28991/cej-2021-03091643
32. Fişne, A., Kuzu, C., Hüdaverdi, T.: Prediction of environmental impacts of quarry blasting operation using fuzzy logic. Environ. Monitor. Assessment **174**(1), 461–470 (2011). https://doi.org/10.1007/s10661-010-1470-z
33. Moonjun, R., Shrestha, D.P., Jetten, V.G.: Fuzzy logic for fine-scale soil mapping: a case study in Thailand. CATENA **190**, 104456 (2020). https://doi.org/10.1016/j.catena.2020.104456

34. Johari, A., Javadi, A.A., Habibagahi, G.: Modelling the mechanical behaviour of unsaturated soils using a genetic algorithm-based neural network. Comput. Geotech. **38**(1), 2–13 (2011). https://doi.org/10.1016/j.compgeo.2010.08.011
35. Hassanlourad, M., Ardakani, A., Kordnaeij, A., Molaabasi, H.: Dry unit weight of compacted soils prediction using GMDH-type neural network. Eur. Phys. J. Plus 132 (2017). https://doi.org/10.1140/epjp/i2017-11623-5
36. Soleimani, S., Rajaei, S., Jiao, P., Sabz, A., Soheilinia, S.: New prediction models for unconfined compressive strength of geopolymer stabilized soil using multi-gen genetic programming. Measurement **113**, 99–107 (2018). https://doi.org/10.1016/j.measurement.2017.08.043
37. Rashidian, V., Hassanlourad, M.: Predicting the shear behavior of cemented and uncemented carbonate sands using a genetic algorithm-based artificial neural network. Geo-tech. Geol. Eng. **31**(4), 1231–1248 (2013). https://doi.org/10.1007/s10706-013-9646-2
38. Beyki Milajerdi, M., Behnamfar, F.: Soil-structure interaction analysis using neural networks optimised by genetic algorithm. Geomech. Geoeng. **17**(5), 1369–1387 (2022). https://doi.org/10.1080/17486025.2021.1940313
39. Cao, Z.-J., Zheng, S., Li, D.-Q., Phoon, K.-K.: Bayesian identification of soil stratigraphy based on soil behaviour type index. Can. Geo-tech. J. **56**(4), 570–586 (2019)
40. Chai, X., Rózsás, Á., Slobbe, A., Teixeira, A.: Probabilistic parameter estimation and reliability assessment of a simulated sheet pile wall system. Comput. Geotech. **142**, 104567 (2022). https://doi.org/10.1016/j.compgeo.2021.104567
41. Aouadj, A., Bouafia, A.: CPT-based method using hybrid artificial neural network and mathematical model to predict the load-settlement behaviour of shallow foundations. Geomech. Geoeng. **17**(1), 321–333 (2022). https://doi.org/10.1080/17486025.2020.1755459
42. Millán, M.A., Galindo, R., Alencar, A.: Application of artificial neural networks for predicting the bearing capacity of shallow foundations on rock masses. Rock Mech. Rock Eng. **54**(9), 5071–5094 (2021). https://doi.org/10.1007/s00603-021-02549-1
43. Jebur, A.A., Atherton, W., Al Khaddar, R.M., Loffill, E.: Artificial neural network (ANN) approach for modelling of pile settlement of open-ended steel piles subjected to compression load. Eur. J. Environ. Civil Eng. **25**(3), 429–451 (2021). https://doi.org/10.1080/19648189.2018.1531269
44. Sethy, B.P., Patra, C., Das, B.M., Sobhan, K.: Prediction of ultimate bearing capacity of circular foundation on sand layer of limited thickness using artificial neural network. Int. J. Geotech. Eng. **15**(10), 1252–1267 (2021). https://doi.org/10.1080/19386362.2019.1645437
45. Gnananandarao, T., Khatri, V.N., Dutta, R.K.: Bearing capacity and settlement prediction of multi-edge skirted footings resting on sand. Ingeniería e Investigación **40**(3), 9–21 (2020). https://doi.org/10.15446/ing.investig.v40n3.83170
46. Pramanik, R., Baidya, D.K., Dhang, N.: Reliability analysis for bearing capacity of surface strip footing using fuzzy finite element method. Geomech. Geoeng. **15**(1), 29–41 (2020). https://doi.org/10.1080/17486025.2019.1601268
47. Moayedi, H., Rezaei, A.: The feasibility of PSO–ANFIS in estimating bearing capacity of strip foundations rested on cohesionless slope. Neural Comput. Appl. **33**(9), 4165–4177 (2021). https://doi.org/10.1007/s00521-020-05231-9
48. Mohammed, M., Sharafati, A., Al-Ansari, N., Yaseen, Z.M.: Shallow foundation settlement quantification: application of hybridized adaptive neuro-fuzzy inference system model. Adv. Civil Eng. **2020**, e7381617 (2020). https://doi.org/10.1155/2020/7381617
49. Jahed Armaghani, D., Harandizadeh, H., Momeni, E.: Load carrying capacity assessment of thin-walled foundations: an ANFIS–PNN model optimized by genetic algorithm. Eng. Comput. **38**(5), 4073–4095 (2022). https://doi.org/10.1007/s00366-021-01380-0
50. Liu, L., Moayedi, H., Rashid, A.S.A., Rahman, S.S.A., Nguyen, H.: Optimizing an ANN model with genetic algorithm (GA) predicting load-settlement behaviours of eco-friendly raft-pile foundation (ERP) system. Eng. Comput. **36**(1), 421–433 (2020). https://doi.org/10.1007/s00366-019-00767-4

51. Murlidhar, B.R., Sinha, R.K., Mohamad, E.T., Sonkar, R., Khorami, M.: The effects of particle swarm optimisation and genetic algorithm on ANN results in predicting pile bearing capacity. Int. J. Hydromechatronics **3**(1), 69–87 (2020). https://doi.org/10.1504/IJHM.2020.105484
52. Zhang, J., Hu, J., Li, X., Li, J.: Bayesian network based machine learning for design of pile foundations. Autom. Construct. **118**, 103295 (2020). https://doi.org/10.1016/j.autcon.2020.103295
53. Christodoulou, P., Pantelidis, L.: Reducing statistical uncertainty in elastic settlement analysis of shallow foundations relying on targeted field investigation: a random field approach. Geosciences **10**(1), Article 1 (2020). https://doi.org/10.3390/geosciences10010020
54. Snieder, E., Shakir, R., Khan, U.T.: A comprehensive comparison of four input variable selection methods for artificial neural network flow forecasting models. J. Hydrol. **583**, 124299 (2020). https://doi.org/10.1016/j.jhydrol.2019.124299
55. Hadiyan, P.P., Moeini, R., Ehsanzadeh, E.: Application of static and dynamic artificial neural networks for forecasting inflow discharges, case study: Sefidroud Dam reservoir. Sustain. Comput.: Informatics Syst. **27**, 100401 (2020). https://doi.org/10.1016/j.suscom.2020.100401
56. Dtissibe, F.Y., Ari, A.A.A., Titouna, C., Thiare, O., Gueroui, A.M.: Flood forecasting based on an artificial neural network scheme. Nat. Hazards **104**(2), 1211–1237 (2020). https://doi.org/10.1007/s11069-020-04211-5
57. Poonia, V., Tiwari, H.L.: Rainfall-runoff modeling for the Hoshangabad Basin of Narmada River using artificial neural network. Arab. J. Geosci. **13**(18), 944 (2020). https://doi.org/10.1007/s12517-020-05930-6
58. Ali, S., Shahbaz, M.: Streamflow forecasting by modeling the rainfall–streamflow relationship using artificial neural networks. Model. Earth Syst. Environ. **6**(3), 1645–1656 (2020). https://doi.org/10.1007/s40808-020-00780-3
59. Perera, E.D.P., Lahat, L.: Fuzzy logic based flood forecasting model for the Kelantan River basin, Malaysia. J. Hydro-Environ. Res. **9**(4), 542–553 (2015). https://doi.org/10.1016/j.jher.2014.12.001
60. Tabbussum, R., Dar, A.Q.: Comparison of fuzzy inference algorithms for stream flow prediction. Neural Comput. Appl. **33**(5), 1643–1653 (2021). https://doi.org/10.1007/s00521-020-05098-w
61. Jayawardena, A.W., Perera, E.D.P., Zhu, B., Amarasekara, J.D., Vereivalu, V.: A comparative study of fuzzy logic systems approach for river discharge prediction. J. Hydrol. **514**, 85–101 (2014). https://doi.org/10.1016/j.jhydrol.2014.03.064
62. Puttinaovarat, S., Horkaew, P.: Flood forecasting system based on integrated big and crowdsource data by using machine learning techniques. IEEE Access **8**, 1–1 (2020). https://doi.org/10.1109/ACCESS.2019.2963819
63. Nguyen, A.D., Le Nguyen, P., Vu, V.H., Pham, Q.V., Nguyen, V.H., Nguyen, M.H., Nguyen, T.H., Nguyen, K.: Accurate discharge and water level forecasting using ensemble learning with genetic algorithm and singular spectrum analysis-based denoising. Sci. Rep. **12**(1), Article 1 (2022). https://doi.org/10.1038/s41598-022-22057-8
64. Young, C.-C., Liu, W.-C., Chung, C.-E.: Genetic algorithm and fuzzy neural networks combined with the hydrological modeling system for forecasting watershed runoff discharge. Neural Comput. Appl. **26** (2015). https://doi.org/10.1007/s00521-015-1832-0
65. Chen, S.-T., Yu, P.-S.: Real-time probabilistic forecasting of flood stages. J. Hydrol. **340**(1), 63–77 (2007). https://doi.org/10.1016/j.jhydrol.2007.04.008
66. Garrote, L., Molina, M., Mediero, L.: Probabilistic forecasts using Bayesian networks calibrated with deterministic rainfall-runoff models. In: Extreme Hydrological Events: New Concepts for Security (pp. 173–183). Springer (2006)
67. Aldhyani, T.H.H., Al-Yaari, M., Alkahtani, H., Maashi, M.: [Retracted] water quality prediction using artificial intelligence algorithms. Appl. Bionics Biomech. **2020**, e6659314 (2020). https://doi.org/10.1155/2020/6659314
68. Hassan, M.M., Hassan, M.M., Akter, L., Rahman, M.M., Zaman, S., Hasib, K.M., Jahan, N., Smrity, R.N., Farhana, J., Raihan, M., Mollick, S.: Efficient prediction of Water Quality Index (WQI) using machine learning algorithms. Hum.-Centric Intell. Syst. **1**(3–4), 86 (2021). https://doi.org/10.2991/hcis.k.211203.001

69. Noori, N., Kalin, L., Isik, S.: Water quality prediction using SWAT-ANN coupled approach. J. Hydrol. **590**, 125220 (2020)
70. Chaudhary, J.K., et al.: A comparative study of fuzzy logic and WQI for groundwater quality assessment. Procedia Comput. Sci. **171**, 1194–1203 (2020)
71. Nayak, J.G., Patil, L.G., Patki, V.K.: Development of water quality index for Godavari River (India) based on fuzzy inference system. Groundwater Sustain. Develop. **10**, 100350 (2020). https://doi.org/10.1016/j.gsd.2020.100350
72. Jha, M.K., Shekhar, A., Jenifer, M.A.: Assessing groundwater quality for drinking water supply using hybrid fuzzy-GIS-based water quality index. Water Res. **179**, 115867 (2020). https://doi.org/10.1016/j.watres.2020.115867
73. Mallik, S., Mishra, U., Paul, N.: Groundwater suitability analysis for drinking using GIS based fuzzy logic. Ecol. Indicators **121**, 107179 (2021). https://doi.org/10.1016/j.ecolind.2020.107179
74. Ghorbani, M.K., Afshar, A., Hamidifar, H.: River water quality management using a fuzzy optimization model and the NSFWQI index. Water SA **47** (2021). https://doi.org/10.17159/wsa/2021.v47.i1.9444
75. Liu, S., Butler, D., Brazier, R., Heathwaite, L., Khu, S.-T.: Using genetic algorithms to calibrate a water quality model. Sci. Total Environ. **374**(2), 260–272 (2007). https://doi.org/10.1016/j.scitotenv.2006.12.042
76. Yang, L., Zhao, X., Peng, S., Li, X.: Water quality assessment analysis by using combination of Bayesian and genetic algorithm approach in an urban lake, China. Ecol. Model. **339**, 77–88 (2016). https://doi.org/10.1016/j.ecolmodel.2016.08.016
77. Sotomayor, G., Hampel, H., Vázquez, R.F.: Water quality assessment with emphasis in parameter optimisation using pattern recognition methods and genetic algorithm. Water Res. **130**, 353–362 (2018). https://doi.org/10.1016/j.watres.2017.12.010
78. Swain, R., Sahoo, B.: Improving river water quality monitoring using satellite data products and a genetic algorithm processing approach. Sustain. Water Qual. Ecol. **9–10**, 88–114 (2017). https://doi.org/10.1016/j.swaqe.2017.09.001
79. Habiyakare, T., Zhang, N., Feng, C.-H., Ndayisenga, F., Kayiranga, A., Sindikubwabo, C., Muhirwa, F., Shah, S.: The implementation of genetic algorithm for the identification of DNAPL sources. Groundwater Sustain. Develop. **16**, 100707 (2022). https://doi.org/10.1016/j.gsd.2021.100707
80. Peng, Z., Hu, Y., Liu, G., Hu, W., Zhang, H., Gao, R.: Calibration and quantifying uncertainty of daily water quality forecasts for large lakes with a Bayesian joint probability modelling approach. Water Res. **185**, 116162 (2020). https://doi.org/10.1016/j.watres.2020.116162
81. Panidhapu, A., Li, Z., Aliashrafi, A., Peleato, N.M.: Integration of weather conditions for predicting microbial water quality using Bayesian Belief Networks. Water Res. **170**, 115349 (2020). https://doi.org/10.1016/j.watres.2019.115349
82. Zare Farjoudi, S., Moridi, A., Sarang, A., Lence, B.: Application of probabilistic bankruptcy method in river water quality management. Int. J. Environ. Sci. Technol. **18**. https://doi.org/10.1007/s13762-020-03046-8
83. Zhou, Y.: Real-time probabilistic forecasting of river water quality under data missing situation: deep learning plus post-processing techniques. J. Hydrol. **589**, 125164 (2020). https://doi.org/10.1016/j.jhydrol.2020.125164
84. Yu, R., Zhang, C.: Early warning of water quality degradation: a copula-based Bayesian network model for highly efficient water quality risk assessment. J. Environ. Manage. **292**, 112749 (2021). https://doi.org/10.1016/j.jenvman.2021.112749
85. Taloba, A.I.: An artificial neural network mechanism for optimizing the water treatment process and desalination process. Alexandria Eng. J. **61**(12), 9287–9295 (2022). https://doi.org/10.1016/j.aej.2022.03.029
86. Abba, S., Usman, A., Danmaraya, Y., Usman, A., Abdullahi, H.: Modeling of water treatment plant performance using artificial neural network: case study Tamburawa Kano-Nigeria. Dutse J. Pure Appl. Sci. (DUJOPAS) **6**(3), 135–14 (2020)

87. Solaimany-Aminabad, M., Maleki, A., Hadi, M.: Application of artificial neural network (ANN) for the prediction of water treatment plant influent characteristics. J. Adv. Environ. Health Res. **1**(2), 89–100 (2013)
88. Tashaouie, H.R., Gholikandi, G.B., Hazrati, H.: Artificial neural network modeling for predict performance of pressure filters in a water treatment plant. Desalin. Water Treatment **39**(1–3), 192–198 (2012). https://doi.org/10.1080/19443994.2012.669175
89. Santín, I., Vilanova, R., Pedret, C., Barbu, M.: New approach for regulation of the internal recirculation flow rate by fuzzy logic in biological wastewater treatments. ISA Trans. **120**, 167–189 (2022). https://doi.org/10.1016/j.isatra.2021.03.028
90. Narayanamoorthy, S., Brainy, J.V., Sulaiman, R., Ferrara, M., Ahmadian, A., Kang, D.: An integrated decision making approach for selecting a sustainable waste water treatment technology. Chemosphere **301**, 134568 (2022). https://doi.org/10.1016/j.chemosphere.2022.134568
91. Godo-Pla, L., Rodríguez, J.J., Suquet, J., Emiliano, P., Valero, F., Poch, M., Monclús, H.: Control of primary disinfection in a drinking water treatment plant based on a fuzzy inference system. Process Saf. Environ. Protect. **145**, 63–70 (2021). https://doi.org/10.1016/j.psep.2020.07.037
92. Okoji, C.N., Okoji, A.I., Ibrahim, M.S., Obinna, O.: Comparative analysis of adaptive neuro-fuzzy inference system (ANFIS) and RSRM models to predict DBP (trihalomethanes) levels in the water treatment plant. Arab. J. Chem. **15**(6), 103794 (2022). https://doi.org/10.1016/j.arabjc.2022.103794
93. Gupta, A.K., Shrivastava, R.K.: Reliability-constrained optimization of water treatment plant design using genetic algorithm. J. Environ. Eng. **136**(3), 326–334 (2010). https://doi.org/10.1061/(ASCE)EE.1943-7870.0000150
94. Swan, R., Bridgeman, J., Sterling, M.: Optimisation of water treatment works performance using genetic algorithms. J. Hydroinformatics **19**(5), 719–733 (2017)
95. Al-Obaidi, M., Li, J.-P., Kara-Zaïtri, C., Mujtaba, I.M.: Optimisation of reverse osmosis based wastewater treatment system for the removal of chlorophenol using genetic algorithms. Chem. Eng. J. **316**, 91–100 (2017)
96. Gino Sophia, S.G., Ceronmani Sharmila, V., Suchitra, S., Sudalai Muthu, T., Pavithra, B.: Water management using genetic algorithm-based machine learning. Soft Comput. **24**(22), 17153–17165 (2020). https://doi.org/10.1007/s00500-020-05009-0
97. Zhu, Z.J.Y., McBean, E.A.: Selection of water treatment processes using Bayesian decision network analyses. J. Environ. Eng. Sci. (2015). https://doi.org/10.1139/s06-030
98. Bertone, E., Rousso, B.Z., Kufeji, D.: A probabilistic decision support tool for prediction and management of rainfall-related poor water quality events for a drinking water treatment plant. J. Environ. Manage. **332**, 117209 (2023). https://doi.org/10.1016/j.jenvman.2022.117209
99. Flores-Alsina, X., Rodríguez-Roda, I., Sin, G., Gernaey, K.V.: Multi-criteria evaluation of wastewater treatment plant control strategies under uncertainty. Water Res. **42**(17), 4485–4497 (2008). https://doi.org/10.1016/j.watres.2008.05.029
100. Marović, I., Androjić, I., Jajac, N., Hanák, T.: Urban road infrastructure maintenance planning with application of neural networks. Complexity **2018**, 1–10 (2018). https://doi.org/10.1155/2018/5160417
101. Jajac, N., Marović, I., Hanák, T.: Decision support for management of urban transport projects. Gradevinar **67**(2), 131–141 (2015)
102. Abu-Lebdeh, G., Chen, H., Ghanim, M.: Improving performance of genetic algorithms for transportation systems: case of parallel genetic algorithms. J. Infrastruct. Syst. **22**(4), A4014002 (2016). https://doi.org/10.1061/(ASCE)IS.1943-555X.0000206
103. Efthymiou, D., Chrysostomou, K., Morfoulaki, M., et al.: Electric vehicles charging infrastructure location: a genetic algorithm approach. Eur. Transp. Res. Rev. **9**, 27 (2017). https://doi.org/10.1007/s12544-017-0239-7
104. Inti, S., Tandon, V.: Application of fuzzy preference–analytic hierarchy process logic in evaluating sustainability of transportation infrastructure requiring multicriteria decision making. J. Infrastruct. Syst. **23**(4), 04017014 (2017). https://doi.org/10.1061/(ASCE)IS.1943-555X.0000373

105. Abeysekara, B.: Application of fuzzy set theory to evaluate large scale transport infrastructure risk assessment and application of best practices for risk management. In: 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, Singapore, pp. 385–389 (2020). https://doi.org/10.1109/IEEM45057.2020.9309957
106. Love, P.E., Ahiaga-Dagbui, D.D., Irani, Z.: Cost overruns in transportation infrastructure projects: sowing the seeds for a probabilistic theory of causation. Transp. Res. Part A: Policy Pract. **92**, 184–194 (2016). https://doi.org/10.1016/j.tra.2016.08.007
107. Van Noortwijk, J., Frangopol, D.M.: Two probabilistic life-cycle maintenance models for deteriorating civil infrastructures. Probab. Eng. Mech. **19**(4), 345–359 (2004). https://doi.org/10.1016/j.probengmech.2004.03.002
108. Kumar, K., Parida, M., Katiyar, V.K.: Short term traffic flow prediction in heterogeneous condition using artificial neural network. Transport **30**(4), 397–405 (2015). https://doi.org/10.3846/16484142.2013.818057
109. Teklu, F., Sumalee, A., Watling, D.: A genetic algorithm approach for optimizing traffic control signals considering routing. Comput.-Aided Civil Infrastruct. Eng. **22**, 31–43 (2007). https://doi.org/10.1111/j.1467-8667.2006.00468.x
110. Mesbah, M., Sarvi, M., Currie, G.: Optimization of transit priority in the transportation network using a genetic algorithm. IEEE Trans. Intell. Transp. Syst. **12**(3), 908–919. (2011). https://doi.org/10.1109/TITS.2011.2144974
111. Peter, A., Zachariah, B., Damuut, L.P., Abdulkadir, S.: Efficient traffic control system using fuzzy logic with priority. In: Misra, S., Muhammad-Bello, B. (eds.) Information and Communication Technology and Applications. ICTA 2020. Communications in Computer and Information Science, vol. 1350. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69143-1_50
112. Jabari, S.E., Liu, H.: A stochastic model of traffic flow: theoretical foundations. Transp. Res. Part B: Methodol. **46**(1), 156–174 (2012). https://doi.org/10.1016/j.trb.2011.09.006
113. Huang, Z.D., Liu, X.J., Huang, C.C., Shen, J.W.: A GIS-based framework for bus network optimization using genetic algorithm. Ann. GIS **16**(3), 185–194 (2010). https://doi.org/10.1080/19475683.2010.513152
114. Cipriani, E., Gori, S., Petrelli, M.: Transit network design: a procedure and an application to a large urban area. Transp. Res. Part C: Emerg. Technol. **20**(1), 3–14 (2012). ISSN 0968-090X. https://doi.org/10.1016/j.trc.2010.09.003
115. An-Hu, R., Bai, J., Ma, J.: Research on intelligent transportation system based on fuzzy neural network. IOP Conf. Ser. Mater. Sci. Eng. **768**(6), 062114 (2020). https://doi.org/10.1088/1757-899x/768/6/062114
116. Koukol, M., ZajíČková, L., Marek, L., Tuček, P.: Fuzzy logic in traffic engineering: a review on signal control. Math. Probl. Eng. 1–14 (2015). https://doi.org/10.1155/2015/979160
117. Deng, W.-J., Chen, W.-C., Pei, W.: Back-propagation neural network based importance–performance analysis for determining critical service attributes. Exp. Syst. Appl. **34**(2), 1115–1125 (2008). ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2006.12.016
118. Aryanti, R., Saryoko, A., Junaidi, A., Marlina, S., Nurmalia, L.: Comparing classification algorithm with genetic algorithm in public transport analysis. In: Journal of Physics: Conference Series (Vol. 1641, No. 1, p. 012017). IOP Publishing (2020). https://doi.org/10.1088/1742-6596/1641/1/012017
119. Şerban, A.: The use of the genetic algorithms for optimizing public transport schedules in congested urban areas. In: IOP Conference Series: Materials Science and Engineering (Vol. 1037, No. 1, p. 012062). IOP Publishing (2021). https://doi.org/10.1088/1757-899X/1037/1/012062
120. Naumov, V., Zhamanbayev, B., Agabekova, D., Zhanbirov, Z., Таран, I.O.: Fuzzy-logic approach to estimate the passengers' preference when choosing a bus line within the public transport system. Komunikácie **23**(3), A150–A157 (2021). https://doi.org/10.26552/com.c.2021.3.a150-a157

121. Klück, F., Zimmermann, M., Wotawa, F., Nica, M.: Genetic algorithm-based test parameter optimization for ADAS system testing. In: 2019 IEEE 19th international conference on software quality, reliability and security (QRS), Sofia, Bulgaria, pp. 418–425 (2019). https://doi.org/10.1109/QRS.2019.00058
122. Al-Saadi, Z., Van Phan, D., Moradi Amani, A., Fayyazi, M., Sadat Sajjadi, S., Ba Pham, D., Khayyam, H.: Intelligent driver assistance and energy management systems of hybrid electric autonomous vehicles. Sustainability **14**(15), 9378 (2022). https://doi.org/10.3390/su14159378
123. Ghahroudi, M.R., Sarshar, M.R., Sabzevari, R.: Introducing a sensor network for advanced driver assistance systems using fuzzy logic and sensor data fusion techniques. Ad Hoc Sens. Wirel. Netw. **8**(1–2), 35–55 (2009)
124. Alimissis, A., Philippopoulos, K., Tzanis, C.G., Deligiorgi, D.: Spatial estimation of urban air pollution with the use of artificial neural network models. Atmos. Environ. **191**, 205–213 (2018). https://doi.org/10.1016/j.atmosenv.2018.07.058
125. Relvas, H., Miranda, A.I.: An urban air quality modeling system to support decision-making: design and implementation. Air Qual. Atmos. Health **11**(7), 815–824 (2018). https://doi.org/10.1007/s11869-018-0587-z
126. Li, J., Zhang, H., Luo, Y., Deng, X., Grieneisen, M.L., Yang, F., Di, B., Zhan, Y.: Stepwise genetic algorithm for adaptive management: application to air quality monitoring network optimization. Atmos. Environ. **215**, 116894 (2019). https://doi.org/10.1016/j.atmosenv.2019.116894
127. Ma, X., Zhang, F.: A Genetic Algorithm Based Stochastic Programming Model for Air Quality Management. IOS Press (2002). https://content.iospress.com/articles/journal-of-environmental-sciences/jes14-3-13
128. Javid, A., Gharibi, H., Sowlat, M.H.: Towards the Application of Fuzzy Logic for Developing a Novel Indoor Air Quality Index (FIAQI). PubMed Central (PMC) (2016). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4841875/
129. Sasaki, S., Comber, A., Suzuki, H., Brunsdon, C.: Using genetic algorithms to optimise current and future health planning—the example of ambulance locations. Int. J. Health Geograph. **9**(1) (2010). https://doi.org/10.1186/1476-072x-9-4
130. Gül, M., Ak, M.F., Güneri, A.F.: Occupational health and safety risk assessment in hospitals: a case study using two-stage fuzzy multi-criteria approach. Hum. Ecol. Risk Assess. **23**(2), 187–202 (2016). https://doi.org/10.1080/10807039.2016.1234363
131. Wang, X., Zhang, M., Zhu, J., Geng, S.: Spectral prediction of *Phytophthora infestans* infection on tomatoes using artificial neural network (ANN). Int. J. Remote Sens. **29**(6), 1693–1706 (2008). https://doi.org/10.1080/01431160701281007
132. Laureano-Rosario, A.E., Duncan, A.P., Méndez-Lázaro, P., Garcia-Rejon, J.E., Gómez-Carro, S., Farfán-Ale, J.A., Savić, D., Müller-Karger, F.E.: Application of artificial neural networks for dengue fever outbreak predictions in the northwest coast of Yucatan, Mexico and San Juan, Puerto Rico. Trop. Med. Infect. Dis. **3**(1), 5 (2018). https://doi.org/10.3390/tropicalmed3010005
133. Traulsen, I., Krieter, J.: Assessing airborne transmission of foot and mouth disease using fuzzy logic. Exp. Syst. Appl. **39**(5), 5071–5077 (2012). https://doi.org/10.1016/j.eswa.2011.11.032
134. Lefevr, N., Kanavos, A., Gerogiannis, V.C., Iliadis, L., Pintelas, P.: Employing fuzzy logic to analyze the structure of complex biological and epidemic spreading models. Mathematics **9**(9), 977 (2021). https://doi.org/10.3390/math9090977
135. Monteiro, L.H.A., Gandini, D., Schimit, P.H.T.: The influence of immune individuals in disease spread evaluated by cellular automaton and genetic algorithm. Comput. Methods Programs Biomed. **196**, 105707 (2020). https://doi.org/10.1016/j.cmpb.2020.105707
136. Johnson, P., Vandewater, L., Wilson, W.J., Maruff, P., Savage, G., Graham, P.L., Macaulay, L., Ellis, K.A., Szoeke, C., Martins, R.N., Rowe, C.C., Masters, C.L., Ames, D., Zhang, P.: Genetic algorithm with logistic regression for prediction of progression to Alzheimer's disease. BMC Bioinformatics **15**(S16) (2014). https://doi.org/10.1186/1471-2105-15-s16-s11

137. Zervoudakis, S., Marakakis, E., Kondylakis, H., Goumas, S.: Prediction of COVID-19 infection based on symptoms and social life using machine learning techniques. In: Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA'21), pp. 277–283. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3453892.3462696
138. Phan, T.D., Smart, J.C.R., Capon, S.J., Hadwen, W.L., Sahin, O.: Applications of Bayesian belief networks in water resource management: a systematic review. Environ. Model. Softw. **85**, 98–111 (2016). https://doi.org/10.1016/j.envsoft.2016.08.006
139. Liao, G., He, P., Gao, X., Lin, Z., Huang, C.J., Zhou, W., Deng, O., Xu, C., Deng, L.: Land use optimization of rural production–living–ecological space at different scales based on the BP–ANN and CLUE–S models. Ecol. Ind. **137**, 108710 (2022). https://doi.org/10.1016/j.ecolind.2022.108710
140. Mohaddes, S.A., Ghazali, M., Rahim, K.A., Nasir, M., Kamaid, A.V.: Fuzzy environmental-economic model for land use planning. Am.-Eurasian J. Agric. Environ. Sci. **3**(6), 850– (2008)
141. Huang, J.S., Liew, J.X., Liew, K.M.: Data-driven machine learning approach for exploring and assessing mechanical properties of carbon nanotube-reinforced cement composites. Compos. Struct. **267**, 113917 (2021). https://doi.org/10.1016/j.compstruct.2021.113917
142. Awolusi, T.F., Oke, O.L., Akinkurolere, O.O., Sojobi, A.O., Aluko, O.G.: Performance comparison of neural network training algorithms in the modeling properties of steel fiber reinforced concrete. Heliyon **5**(1), e01115 (2019). https://doi.org/10.1016/j.heliyon.2018.e01115
143. Taneva, A., Atanasova, D., Daskalov, A.: Fuzzy logic control in building automation application. Eng. Proceed. **41**(1), Article 1 (2023). https://doi.org/10.3390/engproc2023041014
144. Azizi, M., Talatahari, S., Sareh, P.: Design optimization of fuzzy controllers in building structures using the crystal structure algorithm (CryStAl). Adv. Eng. Informatics **52**, 101616 (2022). https://doi.org/10.1016/j.aei.2022.101616
145. Zohrabzadeh, A.M.: Fuzzy Logic application in buildings vibration control in civil engineering. Int. J. Eng. Technol. **11**(4), 740–748 (2019). https://doi.org/10.21817/ijet/2019/v11i4/191104066
146. Razavi Tosee, S.V., Faridmehr, I., Nehdi, M.L., Plevris, V., Valerievich, K.A.: Predicting crack width in CFRP-strengthened RC one-way slabs using hybrid grey wolf optimizer neural network model. Buildings **12** (11) (2022). https://doi.org/10.3390/buildings12111870
147. Tung, T.M., Le, D.-H., Babalola, O.E.: Prediction of residual compressive strength of fly ash based concrete exposed to high temperature using GEP. Comput. Concrete **31**(2), Article 2 (2023)
148. Tung, T.M., Babalola, O.E., Le, D.-H.: Evaluation of the post fire mechanical strength properties of recycled aggregate concrete containing GGBS: optimization and prediction using machine learning techniques. Asian J. Civil Eng. **24**(6), 1639–1666 (2023). https://doi.org/10.1007/s42107-023-00593-6

Machine Learning Applications in Structural Engineering

Ayla Ocak, Gebrail Bekdaş, Sinan Melih Nigdeli, and Umit Işıkdağ

Abstract Machine learning is an artificial intelligence method applied to provide a machine with various abilities such as perception, problem-solving, decision-making, and inference that are unique to human intelligence. With this method, the machine is trained using a data set for the problem to be solved and prediction models are created. The resulting model is based on a mathematical relationship that the machine extracts based on the data. This relationship that the machine establishes between data attributes within itself makes it possible to infer new attributes. Machine learning has supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning options. This method, which has a wide range of applications, has been used effectively in solving many structural engineering problems. In this study, the machine learning method and the studies on its development and applications in the field of structural engineering are described. In addition, a structural engineering study using machine learning is presented.

Keywords Machine learning · Structural engineering · Machine learning application · Truss system

A. Ocak · G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University - Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

U. Işıkdağ
Department of Architect Department, Mimar Sinan Fine-Art University, 34427 Beyoglu, Istanbul, Turkey
e-mail: umit.isikdag@msgsu.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_3

1 Introduction

Artificial intelligence is defined as any software created by imitating the functions of the human brain. The ability to imitate may be the most basic function of the brain, such as learning, but it also includes abilities such as making inferences from what is learned, decision-making, analysis, problem-solving, and similar abilities. Artificial intelligence techniques have been developed by dividing artificial intelligence into sub-branches. Machine learning is a method developed as a sub-branch of artificial intelligence. This method involves the machine gaining the ability to make inferences by training it with some data. The subject of education in a more specific area and with large amounts of data has revealed the deep learning method, which is a sub-branch of machine learning.

The idea of artificial intelligence and machine learning has occupied the minds of many scientists throughout history. Studies inspired by the physical structure of the human brain have been carried out. In 1943, McCulloch and Pitts created a system based on the working principle of neurons in the human brain and stated that the logical operators in their model could assimilate information by creating appropriate networks like the network structure of neurons [1–3]. The Second World War had a significant impact on the formation of the idea of artificial intelligence. At that time, a machine used for communication purposes was encrypted using cryptology, and the British mathematician and computer scientist Alan Turing provided the analysis of this machine that changed the course of the war. Post-war, Turing asked, "Can machines think?" He laid the first intellectual foundations of artificial intelligence by working on the question, and in his article published in 1950, he introduced the Turing test, which he developed to determine whether a human being can be differentiated from a machine [4]. This test laid the foundation for the idea of smart machines of the future by questioning the ability of a machine to imitate a human. Later, with the support of a group of researchers, the future of artificial intelligence as a research discipline started at a conference [5].

Machine learning is a method that aims to obtain human-specific experience and make decisions or inferences based on experience, to create artificial intelligence. In its basic logic, it deals with imitating people's problem-solving ability with the knowledge they have acquired over time. In other words, the machine produces some solutions with the learned information. Human beings are naturally curious and acquire knowledge throughout their lives. Machines acquire the ability to solve knowledge-oriented problems by learning the knowledge acquired throughout human life in a short time. This means that people in a profession can work solution-oriented in a faster and more practical way, compared to the experience they have gained over the years in the same profession, and that machines can also be trained to quickly make inferences that are very close to reality with the data they have learned. For example, when a vehicle needs to be repaired, many parts of the vehicle need to be examined in detail to understand where the problem is. However, a master with many years of experience in his profession can listen to the problems that cause the vehicle to require repair and produce a quick solution to where the problem is. The

machine learning method can be successful in diagnosing the problematic area in the vehicle by providing that information as training, just like a repairman produces solutions by gaining experience. The main important element here is the training data. The details of the training data directly affect the machine's ability to make inferences. Machines can be trained with training on any subject. If you want to train a machine in a more specific area, the process used is deep learning. Deep learning, a sub-branch of machine learning, is a method that focuses on training in a more specific field that requires large amounts of data [6].

2 History of Machine Learning

The historical development of machine learning has progressed in a similar range to the development of artificial intelligence. After Turing's work, artificial intelligence and machine learning continued to develop rapidly. In 1951, Marvin Minsky produced the first neurocomputer called "SNARC", which consisted of 40 neurons [7, 8]. The next studies on the modern machine learning method are the research on the machine that acts like a neuron, developed by Frank Rosenblatt between 1957 and 1960 [9–13]. Rosenblatt presented the first example of artificial neural networks, which he called "perceptron" and which are currently included in the field of machine learning [10]. Arthur Samuel, who worked on computer games, designed a self-learning checkers game in 1959 and was the first to use the term machine learning by including it in his article [14–16]. The 60s correspond to the years when machine learning studies gained momentum with a significant increase. In the early 1960s, studies on learning-recognition and testing phases [17–19] were carried out, and in 1964, the probabilistic expression of ML was proposed with the research [13, 20, 21]. In 1967, the nearest neighbor algorithm, which can be considered a pioneer in pattern recognition, was developed [22, 23]. Afterward, a group of Stanford students began working on controlling a vehicle that would travel on the lunar surface from Earth. The tool they developed was structured for visual navigation studies. By 1979, the Stanford Cart with slider had successfully overcome obstacles in a room full of chairs [24]. In 1981, Gerald Dejong introduced Explanation Based learning, which included data analysis in machine learning, and created a rule to remove unimportant lines in the data [25, 26]. Machine learning and human-specific language discovery abilities such as speaking, pronunciation, and learning became a subject of research in the 80s. In 1985, Tery Sejnowski developed the program called "NETtalk" [27] aimed at learning pronunciation by creating networks that mimic the progress of babies in learning and pronunciation of language from birth. The 90s were the years when data analysis for machine learning was taken into consideration and its importance was emphasized. During this period, research and various programs began to be carried out on machine learning and information extraction from large databases. The analysis and results of large amounts of data using machine learning have been studied, and suggestions have been made to increase database techniques [28]. By 1997, the idea of smart machines witnessed a major impact. That year, the computer

called “Deep Blue”, developed by IBM and capable of playing chess, beat the world chess champion of the time, Garry Kasparov, and set an example of how machines can catch up with human intelligence [29]. The twenty-first century has been the century in which many machine learning algorithms emerged. To eliminate the complexity and efficient use of algorithms, Collobert et al. made a machine learning software library available to scientists in 2002 [30]. In 2006, the deep learning method, a sub-branch of machine learning, emerged [31, 32]. In 2011, the computer named “Watson”, developed by IBM to give possible answers by catching the clues in the questions asked, revived the idea of “machines that can think” by beating its competitors in the Jeopardy word game [33]. In the same year, machine learning and “Google Brain” were developed, in which computers train themselves with previous raw data, with little or no knowledge [34]. In 2014, the Facebook research team developed a system called “DeepFace”, which uses deep neural networks to identify faces and can make identification almost at the level that humans can do [35]. In 2015, Microsoft created the “Distributed Machine Learning Toolkit” for the parallel use of distributing multiple computers for machine learning [36]. Technological progress in the field of computing plays a huge role in the development of machine learning. One of the best examples of this is the program called “AlphaGo”, developed by Google in 2016, which learns and imitates the movements of human players, beating a professional Go player [37]. In the past, the low CPU capacity of computers had a significant impact on the speed of machine learning algorithms. While in 1983, it could take more than 31 min for a computer with a CPU with a speed or frequency of 25 MHz to decide on a move in the Go game, today computers with CPUs above 4 GHz can make this decision in less than 12 s [37]. This clearly shows that the speed of development of computers also affects the development of artificial intelligence and machine learning methods. In 2017, eight researchers working at Google created the deep learning architecture called “transformer”, which is described as a modern artificial intelligence initiative [38]. According to the published article, it is stated that the system they developed can successfully produce images, text, and many similar data in a way that imitates humans [39]. This study formed the basis for the development of machine learning technologies such as natural language processing in the following years. One of the first examples of this was the Bert language model, developed in 2018, which is based on the transformer architecture and can perform functions such as answering questions, making inferences from the language, and similar functions [40]. Between 2018 and 2023, OpenAI developed the GPT series with language generation and understanding capabilities [41–43]. In 2021, Open AI presented the “DALL-E” system, which they developed as a text-to-image translation tool based on machine learning [44]. This system has enabled the creation of many works of art based on the text. Today, systems using machine learning techniques continue to develop. A summary of the historical development of machine learning is visually expressed in Fig. 1.

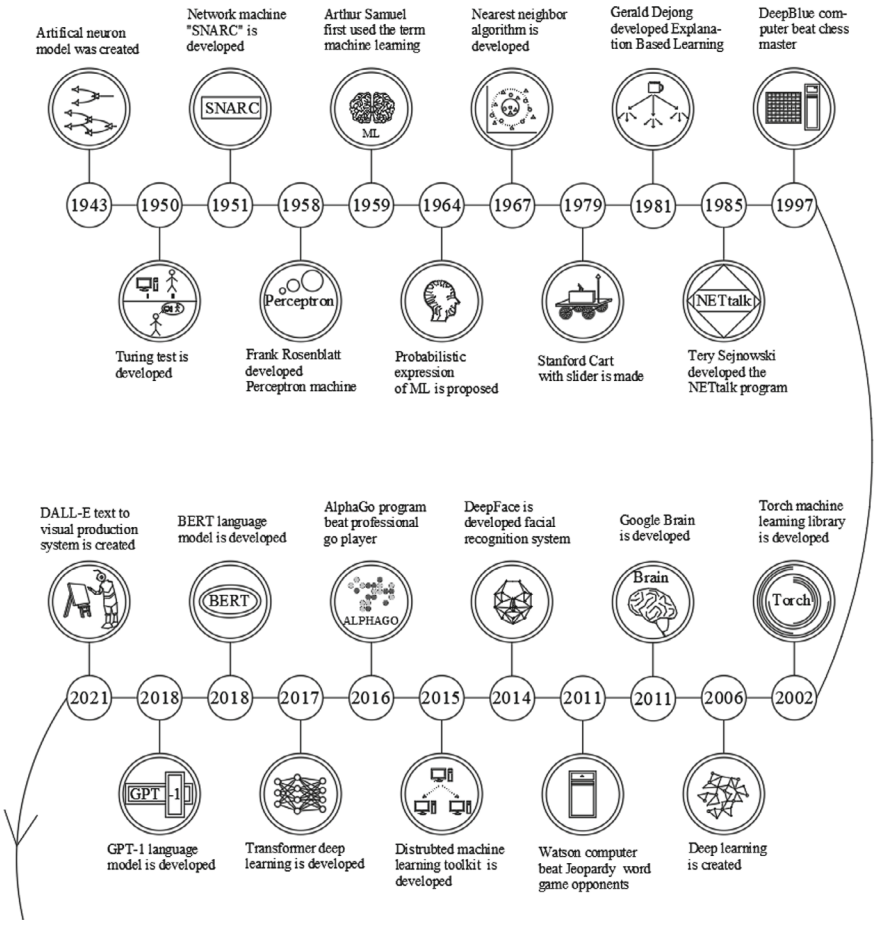


Fig. 1 Timeline of important events in the historical development of machine learning

3 Machine Learning Types

Machine learning is divided into 4 subgroups according to the type of targeted outputs and inputs. These are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning methods. The main factor in determining which method to choose is training data. In a prediction model to be created using the machine learning method, the input parameters in the data and the desired outputs play a big role. If the inputs and outputs in the data are known, by training the machine, output prediction can be made for inputs it has not encountered before. In some cases, a pattern is searched among the inputs only and the output is decided. The aforementioned situations require different machine learning methods. This is one of the reasons why the type of data available and different learning methods

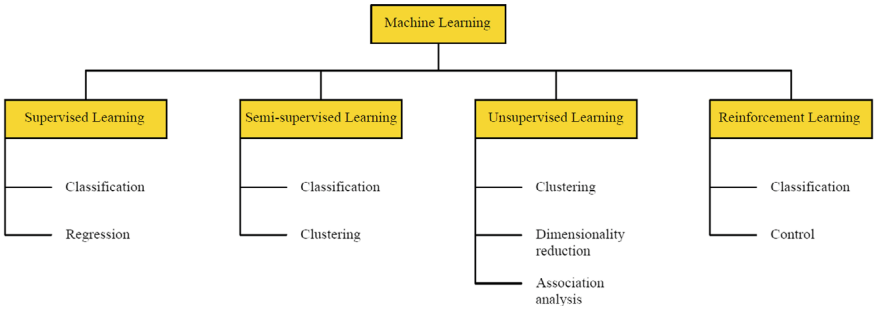


Fig. 2 Types of machine learning

are used. Machine learning types and sub-branches are shown in Fig. 2. Information about machine learning types is presented under the headings below.

3.1 Supervised Learning

Supervised learning is a type of learning in which inputs and outputs are known as training data when creating a machine learning model. Using a sufficient number of input-output pairs in training so that the machine can make accurate predictions generally gives good results [45]. Depending on whether the desired output is numerical or categorical, they are considered and used as classification or regression problems. In supervised learning, the aim is to make inferences from the model created by using labeled data in machine training. The basic principle in supervised learning is that the machine can identify distinctive elements based on data labels. In other words, the main purpose is to create an estimator that can distinguish the features of inputs based on the data label [46]. An example of explaining supervised learning is creating a model with data consisting of various shapes in Fig. 3. Here, the data contains labeled objects with shapes such as circles, squares, and triangles. In supervised learning, the machine learns the labeled data in data using a machine learning algorithm along with their labels and creates a model that explains the label-shape relationship between them. The model that the machine creates based on the information it has learned makes it possible to easily predict the label of the shape when it encounters a new shape.

The example given in Fig. 3 is a classification example where data is categorized using labels. The desired outputs are continuous or discrete, which is why regression or classification algorithms are preferred. In regression problems, the continuous output is called the target value, and in classification problems, the discrete output is called the label [47]. If you want to turn the example in Fig. 3 into a regression problem, using the corner numbers of the shapes as target values instead of data labels can be given as an example. In this case, the machine creates a model that associates each shape with a numerically given number of vertices. When a shape

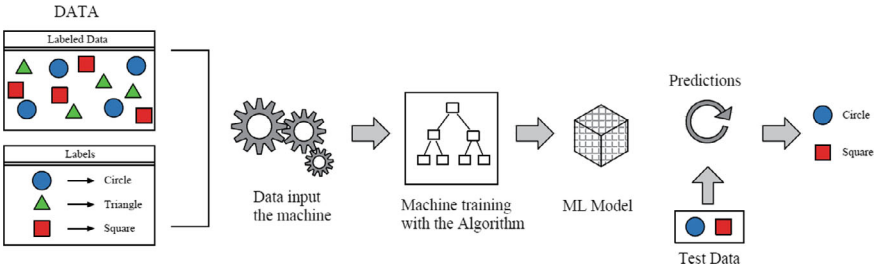


Fig. 3 Supervised learning flowchart

such as a square is shown to the trained machine for testing purposes, the model outputs a prediction of 4, indicating the number of corners.

3.2 Unsupervised Learning

Unsupervised learning is a type of learning in which the machine builds a model from data without data labels. The purpose of this method is to accumulate similar data by taking into account the similarities in the data [48]. It is used as a machine learning method suitable for clustering, dimensionality reduction, and association analysis problems. In unsupervised clustering processes, groups with similar characteristics are identified, discrimination is made and data is clustered. Figure 4 shows a clustering example for Unsupervised learning. Here is data consisting of squares, circles, and triangles. By machine learning data without labels, it clusters similar ones into groups and presents them as output.

Another method used in unsupervised learning is the dimensionality reduction method. The data to be used in machine learning is not always systematic. In some cases, an error in the way data is collected or an unusual situation causes distortions in the data that do not contribute to the results but affect the results. Another reason is that the data is very large. The process of removing data that is not thought to contribute to the machine's prediction performance is called dimensionality reduction. Dimensionality reduction can also be expressed as reducing the size of large amounts of data to facilitate analysis according to their quality and size. Its main purpose is to find the relationship between high-dimensional data and make the

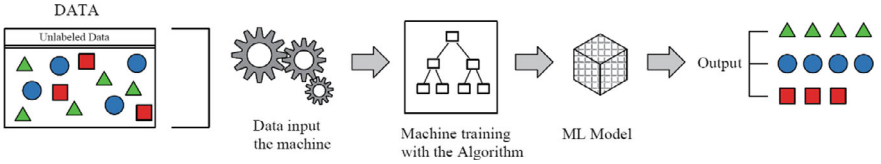


Fig. 4 Unsupervised learning flowchart

data compact [49, 50]. In unsupervised learning, association analysis is performed according to the suitability of the data. The association analysis process analyzes by evaluating the common features found together in the data, just like the products that people prefer to buy together in a grocery store.

3.3 Semi-Supervised Learning

Semi-supervised learning is a machine learning method used in classification and clustering problems. In classification problems, only a part of the data is labeled, and the remaining part is not labeled. The machine creates a model based on the labeled data and associates the unlabeled data. It tries to suggest the closest result to reality when estimating the unlabeled data it encounters. Figure 5 shows a flowchart summarizing semi-supervised learning. Only some of the various geometric shapes are labeled in the figure. Using semi-supervised learning, the machine can identify the square with unlabeled data by creating a model that relates the labeled circle and triangle shapes to the unlabeled square shape.

Labeling data is a process that requires care and time. Especially in large data sets, labeling all data is quite laborious. In such cases, semi-supervised learning is preferred. Semi-supervised learning is a system that processes both supervised and unsupervised learning together. It is recommended to choose a smaller number of labeled data to achieve similar performance with supervised learning [51]. The main reason for this may be that there will be no need for semi-supervised learning since supervised learning algorithms function adequately if the number of labeled data is large. In semi-supervised learning studies, the unlabeled data must be selected in a large amount and the labeled data in a small amount [52]. In this type of learning, studies can be carried out on classification and clustering problems. Clustering is a method for discovering similarities and groups in data. In the semi-supervised learning method, both labeled and unlabeled data are used when clustering [53, 54].

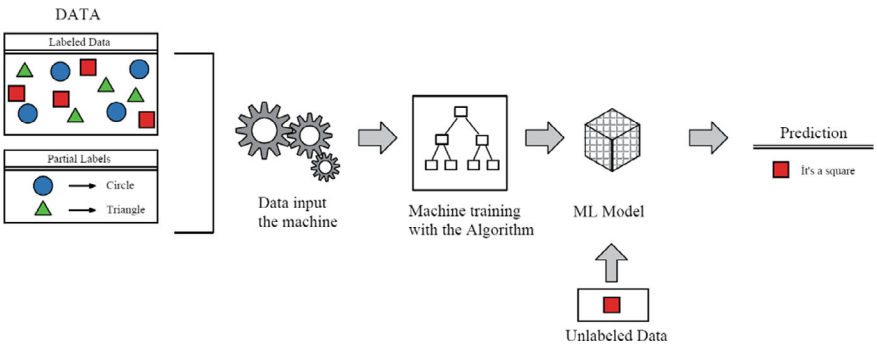


Fig. 5 Semi-supervised learning flowchart

3.4 *Reinforcement Learning*

Reinforcement learning is a machine learning method that operates according to reward-punishment concepts generally preferred in robotics and computer games. In this learning, a student called an agent is used. As a result of his actions, the agent receives rewards or punishments depending on the accuracy of his actions. In reinforcement learning, the agent learns a strategy to maximize reward without human assistance. [55]. Studies on clustering and control problems can be carried out with the reinforcement learning method. When learned from experience, it is categorized as model-independent or model-based [56]. Figure 6 shows the flow chart summarizing the reinforcement learning method. Here, unlabeled data is presented to the machine, as in unsupervised learning. The agent selects these unlabeled data by trial and error and tries to group similar ones. The process it performs is the clustering method. It tries to group similar data and perform clustering, as in unsupervised learning. Output values resulting from machine learning are presented in groups.

Studies on computer games stand out in control studies with reinforcement learning. An example of this is a computer programmed to make chess game moves. As the number of games it plays increases, the computer sees more moves and can record wins and losses as a result of subsequent moves in its memory. Although this is something a person can do, it is a difficult process to remember for a long time. In learning reinforcement, just like a chess master gains experience over the years and learns from his wrong moves, machines also use their experiences to gain knowledge. Here, the reward-punishment relationship can be evaluated as winning and losing. The computer plays each move according to the probability of winning in subsequent moves. Reinforcement learning is the reason why the program called Alpha Go [37] beats the master player in the Go game.

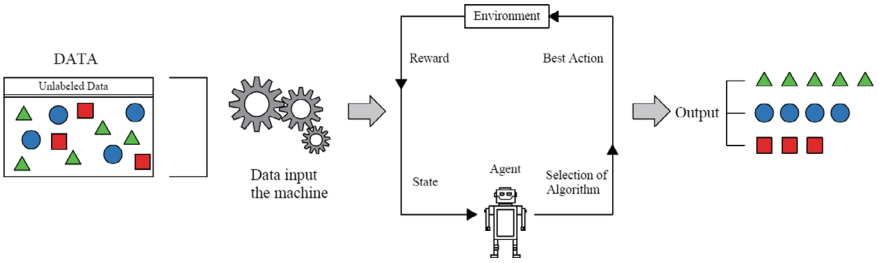


Fig. 6 Reinforcement learning flowchart

4 Machine Learning in Structural Engineering

Machine learning is an effective method that has been used in many studies in all fields of science. Thanks to technology, machines have managed to systematically enter everywhere where people are. Civil engineering is one of the engineering fields where machine learning is frequently used. It is divided into sub-branches such as structure, mechanic, geotechnical, material, construction management, transportation, and hydraulic. Advances in machine learning have increased the use of learning techniques in a variety of civil engineering problems. Looking at the machine learning studies carried out in the field of civil engineering in recent years from a broad perspective, some of the topics that researchers are interested in are as follows: structural damage estimation of a reinforced concrete frame, mechanical properties estimation of composite designs, shear strength estimation in beam and column junctions, concrete compressive strength estimation, concrete electrical resistance estimation, building energy consumption estimation, precipitation estimation, dam hydropower generation estimation, construction labor efficiency estimation, construction cost estimation, seismic response estimation, classification of accident causes in the construction site, classification of concrete cracks, classification of building seismic damage detection and leakage in the water network detection [57–73]. When the studies are examined specifically in structural engineering, they focus on strength estimation of structural system elements, determination of structural design features, seismic response estimation, and similar issues. Following the development of machine learning in recent years, studies in structural engineering have also accelerated. The applications of machine learning and its sub-branches in structural engineering in recent years can be summarized as follows: Liu and Liu performed damage detection by using the support vector machine algorithm, taking into account the curvature mode change in determining the damage area of a long-span bridge [74]. Figueiredo et al. used and compared 4 machine learning algorithms with accelerometer recordings obtained in different environmental and operational situations for vibration-based damage detection of a bottom-excited frame structure [75]. Kiu and Tesfamariam used machine learning algorithms for lateral displacement prediction, where the ground slope was taken into account to determine seismic risk, and achieved better prediction performance with the random forest algorithm [76]. Tezcan and Cheng focused on the prediction of the ground motion response spectrum with the support vector regression algorithm used in supervised learning and produced an alternative to traditional prediction models [77]. Marti-Vargas et al. proposed a model that predicts the transfer length of prestressed steel wires using machine learning methods [78]. Naeej et al. studied confidence coefficient estimation using model trees in reinforced concrete columns [79]. Park et al. used machine learning to understand the changes in wind turbine structures due to the wind field and the wind load effect and focused on predicting the response of the structure to the wind field [80]. Jain et al. used a support vector regression algorithm for sensor-based estimation of energy consumption of occupancy buildings [81]. Jeon et al. used assembly test data of reinforced concrete beam-column connection areas and proposed models

for shear strength [82]. Butcher et al. used the electromagnetic anomaly detection technique of reinforcing steel for flaw detection in reinforced concrete structures and examined neural network approaches to automate data analysis [83]. Behnood et al. created models for predicting tensile strength in steel fiber reinforced and steel-free concrete using model tree, nonlinear regression, artificial neural network, and support vector machine algorithms, and compared the algorithms according to the superiority of the models [84]. Geiß et al. proposed supervised machine learning classification in predicting building type using multi-sensor remote sensing data of seismic buildings [85]. Gonzalez and Karoumi used bridge vibration and load data to determine whether bridges are healthy or damaged and proposed a damage detection system that classifies with machine learning methods [86]. Alimoradi and Beck used machine learning to analyze seismic hazards using strong ground motion data, evaluated the effectiveness of the method with examples, and discussed its advantages and disadvantages [87]. Santos et al. produced a time series using data produced through accelerometers for damage detection from a building sample in a laboratory environment and compared the kernel-based algorithms they used with approaches in the literature [88]. Diez et al. proposed the clustering approach, an unsupervised machine learning method, for the classification of vibrations caused by vehicles and the detection of damaged areas on bridges and investigated its applicability [89]. To examine the deterioration of bridges in reinforced concrete, Kaur et al. presented an approach that classifies and analyzes the deteriorated areas with machine learning methods, using a radar that detects areas where weak signals come from the irons in the deck [90]. Vu and Hoang studied the drilling-shear capacity prediction in fiber-reinforced polymer floors using a hybrid of least squares support vector machine and firefly algorithms [91]. Rafiei and Adeli presented a model that classifies building health by using building response signals taken from different floors with the neural dynamics algorithm they developed for damage detection of large high-rise buildings [92]. Gui et al. created optimized support vector machine models with different metaheuristic algorithms for determining effective features in damage detection of structures and determined the optimization algorithm that makes the methods used more successful in prediction performance [93]. Chatterjee et al. used a neural network to classify structural damage in reinforced concrete structures and optimized the neural network weight coefficients using the genetic algorithm to reduce the root mean square error and maximum error in training the model [94]. Karina et al. proposed a machine learning approach that predicts the tensile strength of worn steel plates using tensile test results, data on the worn surface, and material properties [95]. Modares et al. proposed a convolutional neural network for structural damage prediction, validated their proposed approach, and detected crack damage using composite sandwich panels and concrete bridge examples [96]. Zhang et al. created prediction models with machine learning algorithms to evaluate the safety situation as a result of structural damage after an earthquake and achieved high prediction success in deciding the safety status of the structure [97]. Luo and Paal proposed a machine learning model to predict the hysteresis envelope curves of reinforced concrete columns in a critical state to bending and shear [98]. Geyer and Sundaravelpandian presented a machine learning approach that uses energy

consumption data considering construction components to predict building energy performance [99]. Kiani et al. used various machine learning algorithms to predict fragility curves and structure response in seismic evaluation of structures and examined the sensitivity of the algorithms to the dataset size [100]. Mangalathu et al. used machine learning methods in which bridge characteristics were taken into account for rapid post-earthquake seismic damage assessment of bridges and determined the parameters effective in damage [101]. Using experimental data from infilled frames, Huang and Burton classified the damage modes of the structure with machine learning algorithms [102]. Yücel et al. proposed an artificial neural network model, which is a machine learning method to predict damper parameters with an optimized tuned mass damper using a flower pollination algorithm [103]. Zhu et al. created a convolutional neural network model trained with image data to detect defects in bridges and achieved high-accuracy prediction success by testing the model with different bridge images [104]. Mangalathu et al. built a machine learning model to predict the seismic damage mode using the experimental results of shear walls and achieved a high accuracy in predicting the failure mode [105]. Prayogo et al. presented optimized support vector machine models with metaheuristic symbiotic search algorithms for shear strength prediction in the design of reinforced concrete deep beams [106]. Chen and Chien optimized the weighting matrix of a linear quadratic regulator (LQR) in the active tuned mass damper design with the symbiotic organism algorithm and used machine learning to mimic the control performance [107]. To investigate the load-carrying capacity of steel frame structures, Kim et al. evaluated space frame building models with different story numbers as a machine learning classification and regression problem and compared the algorithm performances [108]. Soize and Orcesi proposed an approach for machine learning-based detection of stiffness changes in structures using experimental data and used bridge data to validate the proposed approach [109]. Hwang et al. investigated the use of machine learning classification and regression methods for collapse prediction based on the response of ductile reinforced concrete frame structures under seismic loads [110]. Kabir et al. used machine learning methods to identify the damage mode of column base plates in steel structures and found effective parameters in determining the damage mode [111]. Kim et al. investigated the identification and clustering of wind pressure patterns in buildings with unsupervised machine learning algorithms [112]. Graciano et al. studied the prediction of the resistance of stainless-steel alloy plate beams under patch loading using machine learning methods [113]. Tong et al. developed a machine learning model that predicts the resistance of thin and I-shaped steel columns at high temperatures using data obtained by the finite element method under axial load [114]. Tang et al. presented a machine learning approach for seismic risk assessment to match buildings' post-earthquake hazard parameters and building parameters [115]. Bekdaş et al. created machine learning models that predict the optimum water tank wall thickness with ensemble learning algorithms using a dataset of optimal design combinations of cylindrical walls [116]. Solhmirzaei et al. developed an approach to predict the flexural capacity of beams made of ultra-high-performance concrete using machine learning techniques [117]. Chen and Feng investigated the effect of low and high-quality data sources on structural response-based machine learning

prediction models and examined the applicability of the method with an example case [118]. Ocak et al. classified the optimum section features with machine learning algorithms using design data obtained through optimization for cost minimization in tubular column and I-section beam examples [119]. Salmi et al. have created models to extract fragility curves for seismic risk assessment of moment-resistant reinforced concrete frame structures with different classification algorithms [120]. Barkhordari and Tehranizadeh studied the prediction of the damage mode of W-shaped steel columns with machine learning algorithms based on experimental data and proposed a model [121]. Naresh et al. have designed a model using vibration data for damage classification for structural health monitoring of steel frame structures with machine learning techniques [122]. Fan created a model with machine learning algorithms for damage detection of reinforced concrete structures from images, and evaluated the damage recognition performance of the model for different scenarios such as lighting that affects image quality [123]. Yu and Zhang proposed a photogrammetric measurement system with a weighted position algorithm based on machine learning to measure displacements caused by various environmental problems in long-span bridges [124]. Using various machine learning algorithms to predict buckling strength using data from experimental and numerical studies for the design of I-section columns made of high-strength steel, Cheng et al. were able to predict the damage load with remarkable accuracy [125]. Ocak et al. developed a model that predicts the damping capacity of seismic isolators designed for structures subject to earthquake excitations using machine learning techniques [126]. Yan et al. used wind tunnel data to identify vortex-induced vibrations in bridges and created a model that distinguishes vortex signals with machine learning clustering algorithms [127].

5 Machine Learning Application for a Truss System

Cage systems are one of the application areas of civil engineering where safety and cost are taken into consideration in design. In the design of these systems, the amount of material, labor, transportation, and application are among the main elements of construction costs. Various optimization methods are used to build truss systems safer and more cost-effectively. In optimization, bar cross-sections are optimized by taking into account the structure boundary conditions and the aim is to reduce the material volume to minimize the structure cost. In the optimization of such systems, metaheuristic algorithms are preferred due to their simple and understandable structure and ease of implementation. Metaheuristic algorithms are algorithms obtained by mathematically transforming various events in nature or the instinctive behaviors of living things [128–132]. There are many studies using metaheuristic algorithms in truss system optimization [133–138]. Optimizing the truss structure design provides cost savings in terms of time and labor. In particular, optimizing building element cross-sections is one of the main factors affecting the cost. In recent years, with the development of artificial intelligence technologies, machine learning methods

have begun to be used to estimate optimal design parameters. Machine learning methods are data-based systems. The first step required for such systems to successfully produce predictive models is the creation of a good database. The optimization process is an effective data generation method to create a database in truss system design. When the studies are examined, it is seen that studies using metaheuristic-based optimization and machine learning methods in truss system designs have become widespread [139–143].

In this study, the cross-section of 3-bar truss system elements was optimized on the Matlab program for various load combinations to minimize the building volume by using the Jaya algorithm, which is a metaheuristic algorithm [144, 145]. A data set for machine learning was created from the optimum parameters obtained from optimization. With the optimum system data produced, cross-sectional prediction models were created using various machine learning algorithms on the Python program [146]. The main purpose of the study is to create a model that predicts the bar sections used in a sample truss system, taking into account only the load and strain states. By comparing prediction models created with different algorithms, the most appropriate estimator was decided.

5.1 Methodology

In this section, information is given about the Jaya algorithm to be used in the optimization of the truss system example, the learning algorithms used for the machine learning model, the metrics to be used in machine learning model evaluation, and the model verification method.

5.1.1 Optimization with Jaya Algorithm

Jaya algorithm is a single-stage metaheuristic algorithm developed by Rao in 2016 [144]. It stands out among metaheuristic algorithms in terms of its understandable structure that can be easily applied to problems and the speed of reaching the optimum solution. In addition, it does not contain any design factors specific to the algorithm and does not require parameter adjustments, which are some of its features that accelerate the solution process. In the optimization process with the Jaya algorithm, firstly, the fixed values of the structure to be optimized, the number of populations, the number of iterations, and the lower and upper limits of the features to be optimized are introduced to the system. Then, the first solutions for the optimization process are generated randomly at the given limit values. The solution generation process is a process that repeats the number of iterations and each new solution matrix is produced by the algorithm equation. There is only one solution equation specific to Jaya. The solution production equation is given in Eq. 1.

$$X_{i,new} = X_{i,j} + rand\left(X_{i,best} - \left|X_{i,j}\right|\right) - rand\left(X_{i,worst} - \left|X_{i,j}\right|\right) \tag{1}$$

In Eq. 1, the best solution in the solution matrix is shown as $X_{i,best}$, the worst solution is $X_{i,worst}$, the new candidate solution is $X_{i,new}$, and the jth solution of the ith variable is shown as $X_{i,j}$.

While each solution is produced, the objective function of the problem is calculated. The produced solutions are compared with the old solutions according to the objective function, and if the new solutions are better than the old solutions, the old ones are updated. This process is repeated for as many iterations as possible. When all iterations are completed, optimum solutions that are most suitable for the objective function are obtained.

5.1.2 Machine Learning Algorithms

In machine learning methods, the learning method is decided according to the type and size of the data. For the truss system example, which is the subject of this study, the system was solved as a regression problem in the field of supervised learning methods. In practice, lasso, ridge, elasticnet, linear regression, support vector machine, and k-nearest neighbor machine learning algorithms were used.

Lasso regression is a machine learning algorithm that is used to reduce data and is effective in predicting rare features in the data even afterward [147]. What the algorithm does is a type of data minimization. It is used effectively to eliminate complexity when there are many input parameters in the data. It aims to determine the necessary input parameters and their regression coefficients to keep the estimation error in the model at the lowest level [148].

Ridge regression [149] is a machine learning algorithm that deals with the relationship between independent variables in the data. It focuses on solving the multicollinearity problem caused by two or more independent variables having high correlation. It identifies the least effective features in the data and reduces the coefficients of these variables [150]. This feature of the algorithm ensures that the model with low-impact variables can produce good predictions and avoid high variance by determining the regression coefficients correctly.

ElasticNet algorithm is a machine learning algorithm that uses Lasso regression to remove low-impact inputs in the data and Ridge regression to reduce coefficients according to their importance in the data [150]. It is recommended to prevent the instability of Lasso regression in cases where the correlation between input parameters in the data is high [151, 152].

The Support Vector Machine algorithm [153] creates a hyperplane in grouping the data when performing regression and creates a line that includes more points by giving high margin limits on this plane. Not choosing margin limits wide enough is a factor that increases the regression error of the algorithm [154]. In the hyperplane, error is minimized by penalizing data outside the margin limits.

Linear Regression is a method that aims to linearly explain the relationship between input values (independent variables) and desired output values (dependent variables) in data. It is assumed that there is a line showing the dependent variables, and a linear relationship is tried to be created by weighting the independent variables

around this line. Dependent variables in the data are expected to be continuous and independent variables to be continuous or discrete [154, 155].

K-Neighbor Nearest: The nearest neighbor algorithm is a machine learning algorithm that calculates the outputs by considering the average or frequency of k observation units determined around the sample. It is suitable for classification and regression problems. In regression problems, it aims to predict by finding the k closest observation samples based on the distance of a randomly selected sample [156]. The selected number of samples "k" is one of the main factors affecting the performance of the algorithm. The higher the number of observation samples around the random sample, "k", the more data is included in the voting to decide the class or value of the sample, and therefore the model can produce more accurate predictions [157]. However, choosing this value too high may reduce the negative effect of the variance as a result of the error caused by randomly selected samples, but may also cause the model to be ignored [158]. The important thing to consider when choosing the optimal K value is to maintain a balance between these limits without over- or under-fitting [158, 159]. After determining the k neighbors at a selected distance from the randomly taken sample, these observations are voted according to the type of problem, and the estimated value of the regression sample is reached according to the class of the sample and the output values of the samples. A categorical example that schematically illustrates the k-nearest neighbor algorithm is given in Fig. 7. K $= 3$ and k $= 10$ observation samples at the specified distance to the selected sample are shown. In deciding the class of a randomly selected sample, the decision is made based on the class of k number of samples.

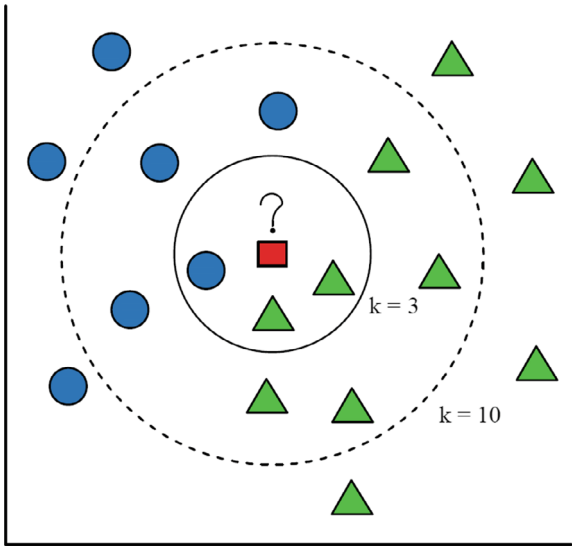


Fig. 7 k-nearest neighbor algorithm example

5.1.3 Model Metrics

Various model metrics are used to evaluate a machine learning model. The regression problem was chosen for the sample study. In regression problems, the R-squared (R^2) score is considered as the main evaluation criterion to evaluate the performance of the model. Apart from this value, some numerical study metrics such as mean absolute error (MAE), mean squared error (MSE), and root mean square error (RMSE) are also used. Equation 2 shows the R2 calculation equation. MAE, MSE, and RMSE metrics are given in Eqs. 3, 4, and 5, respectively.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N} \left(y_i - \overline{y}_i\right)^2} \tag{2}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left|\hat{y}_i - y_i\right| \tag{3}$$

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i - y_i\right)^2 \tag{4}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i - y_i\right)^2} \tag{5}$$

5.1.4 Validation Method of the Model

Various verification methods are used for prediction models obtained by machine learning methods. The purpose here is to verify the prediction performance of the model according to the data allocated for training and testing. The k-fold cross-validation method was used for the sample truss system study. This method is a simple and understandable verification method developed based on repeated use of samples [160, 161]. The basic principle of the method is to divide the data into k folds and separate each fold for training and testing to ensure k model evaluations. For example, selecting k as 5 means dividing the data into 5 parts and separating each 5 parts as training and test data to obtain separate performance scores. The main advantage of the method is to verify the random training and testing phase taken from all data with many folds. As a result, the mean of R-squared values obtained from all folds indicates the overall performance of the model. A summary of the use of the verification method for data divided into 5 folds is shown in Fig. 8.

Fig. 8 Fivefold cross-validation method

5.2 *Numerical Example*

In this study, an optimization process was carried out to reduce the material volume using the Jaya algorithm for a 3-bar truss system example where two different cross-section bars are used. In optimization, bar cross-sectional areas were optimized on the Matlab program [145]. The example used in the study is shown in Fig. 9.

In optimization, bar cross-sectional areas were produced depending on the load (P) and strain (σ) values, design constraints were used, and material volume was minimized. The parameters used in optimization are given in Table 1, and the design constraints and objective functions are given in Table 2.

A data set was prepared by taking optimized bar cross-sectional areas, load, and strain values. 1029 lines of data were created from optimized values for use in machine learning. The first 10 lines of the created data are shown in Table 3. Additionally, values summarizing the data such as the mean, standard deviation, minimum, and maximum values of the data are given in Table 4, the correlation matrix explaining the relationship of the values in the data is given in Table 5, and the visual correlation matrix representation is given in Fig. 10.

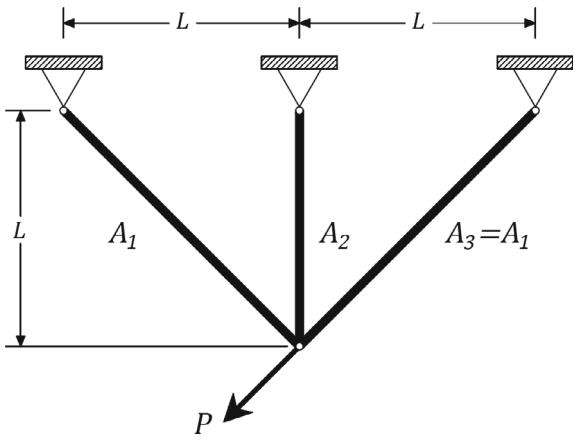


Fig. 9 3-Bar truss system model

Table 1 Optimization parameters and design limits

Symbol	Definition	Value
pn	Population number	15
maxiter	Maximum number of iterations	10,000
A1, A2 min	Maximum cross-sectional area	0
A1, A2 max	Bar length (cm)	1
L	External load (kN)	100
P	Strain (kN/cm^2)	1–40
σ		1–40

Table 2 Design constraint equations and objective functions

Definition	Equation
Design constraints	$g_1 = \frac{\sqrt{2}A_1 + A_2}{\sqrt{2}A_1^2 + 2A_1A_2} P - \sigma \leq 0$
	$g_2 = \frac{A_2}{\sqrt{2}A_1^2 + 2A_1A_2} P - \sigma \leq 0$
	$g_3 = \frac{1}{A_1 + \sqrt{2}A_2} P - \sigma \leq 0$
Objective Function	Minimize $f(A_1, A_2) = (2\sqrt{2}A_1 + A_2)L$ $0 \leq A_1 \leq 1$ and $0 \leq A_2 \leq 1$

Table 3 First 10 rows of the dataset

Strain σ (kN/cm^2)	External LOAD P (kN)	Cross-sectional area A1 (cm^2)	Cross-sectional area A2 (cm^2)
1	1	0.77649	0.44413
2	1	0.39434	0.204116
2	2	0.788836	0.409622
3	1	0.262898	0.136066
3	2	0.538552	0.23852
3	3	0.77759	0.445321
3	4	0.999803	0.709663
4	1	0.197158	0.102092
4	2	0.394341	0.204114
4	3	0.591521	0.306145

5.3 Machine Learning Model

In this study, the 3-bar truss system example was optimized with the Jaya algorithm to give the minimum material volume under different load and strain conditions. A data set was created with 2 different optimized bar cross-sections, load, and strain values. Machine learning was performed on the resulting data set using machine learning algorithms. Since the data used in the study was numerical, it was treated as

Table 4 Descriptive statistics of the dataset

Feature	σ	P	A1	A2
	kN/cm^2	kN	cm^2	cm^2
Minimum variable	1	1	0.0197	0.0102
Maximum variable	40	40	1	0.9975
Mean	26.28	16.76	0.5168	0.2833
Variance	87.89	119.05	0.0849	0.0352
Standard deviation	9.38	10.92	0.2916	0.1878
25%	20	7	0.2629	0.1361
50%	28	15	0.5126	0.2642
75%	34	25	0.7624	0.3867

Table 5 The correlation matrix

	Strain (ST)	External load (P)	Cross-sectional area 1 (A1)	Cross-sectional area 2 (A2)
Strain (ST)	1	0.4184	−0.1553	−0.1754
External load (P)	0.4184	1	0.7801	0.6805
Cross-sectional area 1 (A1)	−0.1553	0.7801	1	0.8864
Cross-sectional area 2 (A2)	−0.1754	0.6805	0.8864	1

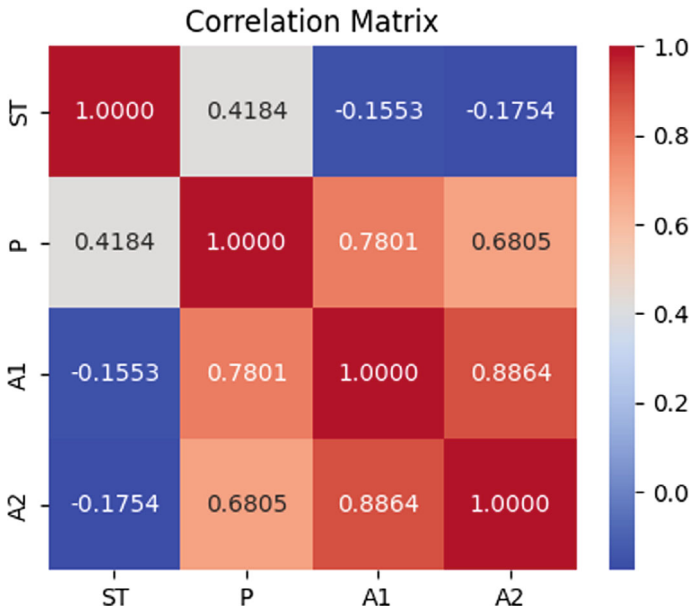


Fig. 10 The correlation matrix

Table 6 Model evaluation results obtained from regression algorithms for the sample data set

Algorithm	MAE	MSE	RMSE	Max-R^2	Mean-R^2
Lasso	0.1462	0.0347	0.1846	0.3994	0.3602
ElasticNet	0.1000	0.0184	0.1348	0.7100	0.6293
Linear regression	0.0670	0.0097	0.0978	0.8285	0.8014
Ridge	0.0668	0.0097	0.0977	0.8741	0.8060
Support vector machine	0.0524	0.0049	0.0671	0.9107	0.8736
K-neighbor	0.0270	0.0034	0.0530	0.9494	0.9210

a regression problem and the 10-fold cross-validation method was used to verify the created machine learning model. Model evaluation metrics (Mean-R^2, MAE, RMSE, MSE, and Max-R^2) of the created model are shown in Table 6. The maximum R-squared score obtained within 10 folds and the mean R^2 scores of 10 folds are shown separately in the table.

5.4 Discussion and Conclusions

In this study, a 3-bar truss system example was optimized with the Jaya algorithm, a metaheuristic algorithm, to reduce the cost by minimizing the material volume. In the optimization, two optimum bar cross-sectional areas were sought for different load and strain combinations. A data set was created to apply machine learning with the optimum cross-sectional features, load, and strain values obtained. The aim here is to create a model that predicts the most suitable bar cross-sections according to the load and strain state. The developed models were validated using the 10-fold cross-validation method. Mean absolute error (MAE) values of the machine learning models obtained in the study are shown in Fig. 11, mean squared error (MSE) values are shown in Fig. 12, and Mean R-squared (R2) scores of 10 folds are shown in Fig. 13.

When the MAE values of the models are examined, it is seen that the lowest Mean absolute error value is 2.7% in the model obtained with the K-nearest neighbor (KNN) algorithm. Support vector machine (SVM), Linear regression (LR), and Ridge regression gave relatively close results. According to the values obtained with Ridge (RR), Lasso, and ElasticNet (EN) algorithms, an MAE value of approximately 6.7% was calculated with the Ridge regression model, while MAE values of 14.6% were obtained with Lasso and 10% with ElastiNnet. Although the ElasticNet algorithm is a method that combines the Ridge and Lasso algorithms, it has a higher MAE value than the Ridge regression algorithm and lower than the Lasso regression. This may be because Lasso regression removes data that has little impact on the data. Ridge regression, on the other hand, reduces the weight coefficients instead of removing low-importance data. Considering the amount of data in the example problem, in the 1029 rows of data, reducing the weights of low-impact inputs rather

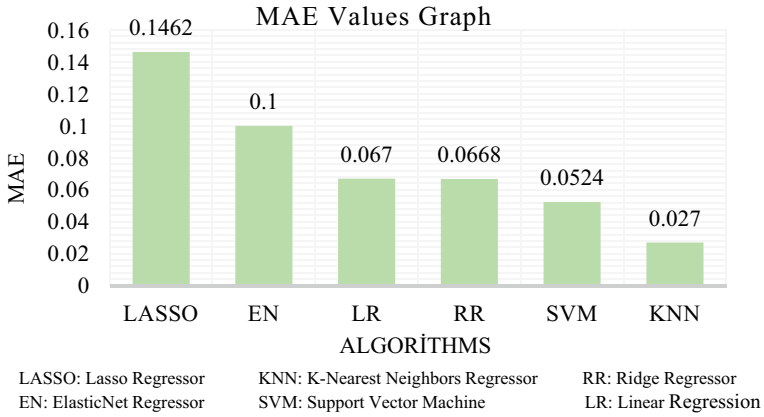


Fig. 11 Graph of MAE values of algorithms

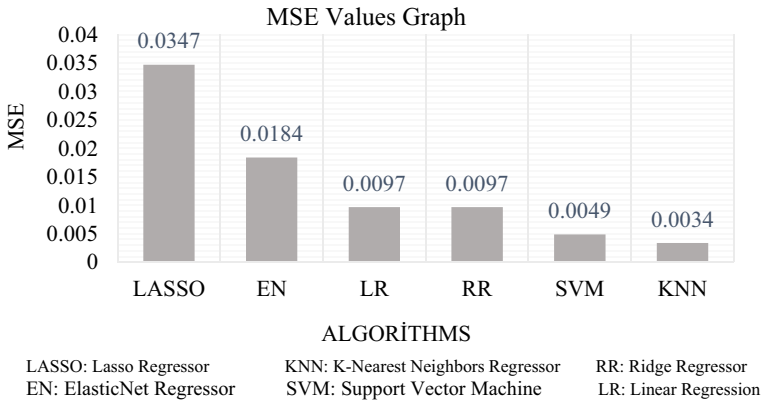


Fig. 12 Graph of MSE values of algorithms

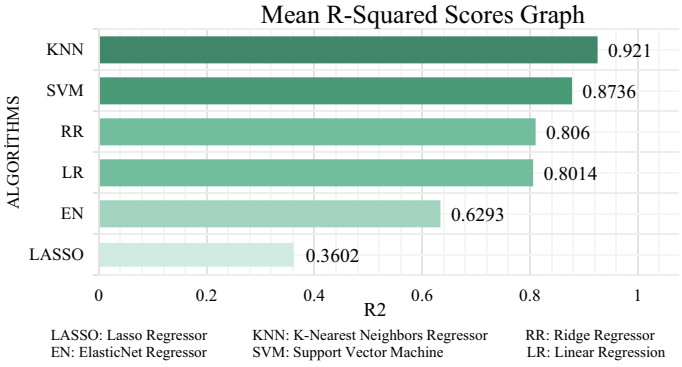


Fig. 13 Graph of R-squared values of algorithms

than removing them is more effective in the model. For this reason, it can be said that the ElasticNet algorithm's inclusion of the Ridge algorithm can slightly improve the negative effect of the Lasso regressor and produce better MAE values.

The mean squared error (MSE) values of the models give similar results to the MAE values. According to the results, the lowest MSE value is seen in the KNN algorithm with a rate of 0.3%.

According to the Mean-R2 scores shown in Fig. 13, the highest prediction success rate is seen in the KNN algorithm with a rate of 92.1%. Prediction models developed with SVM, RR, and LR algorithms showed a performance exceeding 80%. SVM has been the second most successful algorithm. This may be because the SVM algorithm creates a model that includes more points in the data by giving higher margin limits. LR algorithm is based on the linearity relationship established between dependent and independent variables. The reason why the LR algorithm performs similarly well may be that a linear relationship can be established between cross-sectional areas and load and strain. KNN algorithm is an algorithm that produces predictions by determining the distance of the nearest neighbors and the number of observations (k). The highest prediction success in the studied model was seen in this algorithm.

When all the results are examined, it is seen that the cross-section values of a 3-bar simple truss system can be predicted at a level exceeding 92% for data prepared by taking only load and strain values into account. The resulting solution presented a model for the problem. However, it is promising that by increasing the number of data and obtaining optimum design data, various properties of more comprehensive truss systems can be predicted.

6 Summary and Conclusions

In this study, the machine learning method is presented in general, machine learning techniques, the historical development of machine learning, some of the topics covered in civil engineering applications, and a summary of current studies in structural engineering are described. As a result of the literature survey, it has been observed that machine learning methods have been the subject of many studies, especially in the prediction of structural damage. Moreover, it is understood that it is a frequently preferred method for structural health monitoring, determination of seismic fragility and response of structures, estimation of shear and flexural strengths of structural elements, and determination of damage mode. In addition, a simple truss system example was taken and optimized with the Jaya algorithm, a metaheuristic algorithm for use in machine learning methods. The obtained optimum results were converted into a data set for various load and strain combinations. Regression prediction models were created using various machine learning algorithms in the application example. As a result of the developed model, it is seen that machine learning methods have the potential in structural engineering applications to produce optimal prediction models for new designs with optimum design features.

References

1. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**, 115–133 (1943)
2. Russel S.J., Norvig P., et al.:. Artificial intelligence: the gestation of artificial intelligence (1943–1956). In: Artificial Intelligence: A Modern Approach, pp. 16. Prentice-Hall, Inc., (1995). ISBN 0-13-103805-2
3. Prasad, R., Choudhary, P.: State-of-the-art of artificial intelligence. J. Mobile Multimedia **17**(1–3), 427–454 (2021)
4. Turing, A.M.: Computing machinery and intelligence. Mind **59**(236), 433–460 (1950)
5. Moor, J.: The Dartmouth College artificial intelligence conference: the next fifty years. AI Mag. **27**(4), 87–87 (2006)
6. Ocak, A., Nigdeli, S.M., Bekdaş, G., Işıkdağ, Ü.: Artificial intelligence and deep learning in civil engineering. In: Hybrid Metaheuristics in Structural Engineering: Including Machine Learning Applications, pp. 265–288. Springer Nature Switzerland, Cham (2023)
7. Minsky, M.: Neural Nets and the Brain-Model Problem. Ph.D. Dissertation, Princeton University, Princeton, NJ (1954)
8. Poulton, M.M.: A brief history. In: Handbook of Geophysical Exploration: Seismic Exploration, vol. 30, pp. 3–18. Pergamon (2001)
9. Rosenblatt, F.: The Perceptron, A Perceiving and Recognizing Automaton, Project Para Report No. 85-460-1, Cornell Aeronautical Laboratory (CAL) (1957)
10. Rosenblatt, F.: The Perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**, 386–408 (1958)
11. Rosenblatt, F.: Two theorems of statistical separability in the perceptron. In: Symposium of the Mechanisation of Thought Processes. National Physical Laboratory, Teddington, UK, Nov. 1958, Vol I, H.M. Stationery Office, London (1959)
12. Rosenblatt, F.: Perceptron simulation experiments. Proc. Inst. Radio Eng. **18**, 301–309 (1960)
13. Fradkov, A.L.: Early history of machine learning. IFAC-PapersOnLine **53**(2), 1385–1390 (2020)
14. Samuel, A.L.: Some studies in machine learning use the game of checkers. IBM J. Res. Dev. **3**(3), 211–229 (1959)
15. Samuel, A.L.: Machine learning. Technol. Rev. **62**(1), 42–45 (1959)
16. Aydin, Y., Bekdaş, G., Işıkdağ, Ü., Nigdeli, S.M.: The state of art in machine learning applications in civil engineering. In: Hybrid Metaheuristics in Structural Engineering: Including Machine Learning Applications, 147–177 (2023)
17. Widrow, B.: Self-adaptive discrete systems. Theory Self Adapf. Contr. Syst. Proc. IFAC Symp. (1961)
18. Bongard M.M.: Simulation of the recognition process on a digital computing machine. Biophysics 4(2) (1961)
19. Braverman, E.M.: The experiments with training a machine to recognize patterns. Autom. Remote Contr. **23**(3) (1962)
20. Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: Theoretical foundations of the potential function method in the problem of training automata to classify input situations. Autom. Remote Contr. (USSR) **25**(6) (1964)
21. Aizerman, M.A., Braverman, E.M., Rozonocr, L.I.: The probabilistic problem of training automata to recognize patterns and the potential function method. Autom. Remote Contr. (USSR) **25**(9) (1964)
22. Qi, C., Yilmaz, E., Chen, Q.: Background of machine learning. In: Machine Learning Applications in Industrial Solid Ash, pp. 93–130. Elsevier (2024)
23. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
24. https://web.stanford.edu/~learnest/sail/oldcart.html. [Visit Date: 11 May 2024]
25. Minton, S., Zweben, M.: Learning, planning, and scheduling: an overview. Machine Learning Methods for Planning, pp. 1–29 (1993)

26. Alzubi, J., Nayyar, A., Kumar, A.: Machine learning from theory to algorithms: an overview. In: Journal of Physics: Conference Series, vol. 1142, p. 012012. IOP Publishing (2018)
27. Sejnowski, T.: Net talk: a parallel network that learns to read aloud. Complex Syst. **1**, 145–168 (1987)
28. Han, J., Cai, Y., Cercone, N.: Data-driven discovery of quantitative rules in relational databases. IEEE Trans. Knowl. Data Eng. **5**(1), 29–40 (1993)
29. https://www.ibm.com/history/deep-blue. [Visit Date: 11 May 2024]
30. Collobert, R., Bengio, S., Mariéthoz, J.: Torch: a modular machine learning software library (2002)
31. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
32. Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Inform. Process. **3**, e2 (2014)
33. https://www.ibm.com/watson. [Visit Date: 11 May 2024]
34. Helms, M., Ault, S.V., Mao, G., Wang, J.: An overview of google brain and its applications. In: Proceedings of the 2018 International Conference on Big Data and Education, pp. 72–75 (2018)
35. Taigman, Y., Yang, M., Ranzato, M.A., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)
36. https://www.microsoft.com/en-us/research/blog/microsoft-open-sources-distributed-machinelearning-toolkit-for-more-efficient-big-data-research/. Visit Date: 11 May 2024
37. Chen, J.X.: The evolution of computing: AlphaGo. Comput. Sci. Eng. **18**(4), 4–7 (2016)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Adv. Neural Inform. Process. Syst. **30** (2017)
39. https://www.bloomberg.com/opinion/features/2023-07-13/ex-google-scientists-kickstarted-the-generative-ai-era-of-chatgpt-midjourney. [Visit Date: 11 May 2024].
40. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
41. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Visit Date: 11 May 2024
42. Floridi, L., Chiriatti, M.: GPT-3: its nature, scope, limits, and consequences. Mind. Mach. **30**, 681–694 (2020)
43. https://openai.com/index/gpt-4/. Visit Date: 11 May 2024
44. https://web.archive.org/web/20210106082657/https://venturebeat.com/2021/01/05/openai-debuts-dall-e-for-generating-images-from-text/. Visit Date: 11 May 2024
45. Jiang, H.: Machine Learning Fundamentals: A Concise Introduction. Cambridge University Press (2021)
46. Ali, A., Mashwani, W.K.: A supervised machine learning algorithms: applications, challenges, and recommendations. Proc. Pakistan Acad. Sci. A. Phys. Comput. Sci. **60**(4), 1–12 (2023)
47. Deprez, M., Robinson, E.C.: Machine learning basics. In: Machine Learning for Biomedical Applications, pp. 41–65. Academic Press (2024)
48. Chander, S., Vijaya, P.: Unsupervised learning methods for data clustering. In: Artificial Intelligence in Data Mining, pp. 41–64. Academic Press (2021)
49. Niu, F., Zhao, X., Guo, J., Shi, M., Liu, X., Liu, B.: Fast and robust unsupervised dimensionality reduction with adaptive bipartite graphs. Knowl.-Based Syst. **276**, 110680 (2023)
50. Zhen, L., Peng, X., Peng, D.: Local neighborhood embedding for unsupervised nonlinear dimension reduction. J. Softw. **8**(2), 410–417 (2013)
51. Shanthamallu, U.S., Spanias, A.: Semi-supervised learning. In: Machine and Deep Learning Algorithms and Applications. Synthesis Lectures on Signal Processing. Springer, Cham (2022)
52. Shobha, G., Rangaswamy, S.: Chapter 8-Machine Learning Handbook of Statistics. Elsevier (2018)

53. Reddy, Y.C.A.P., Viswanath, P., Reddy, B.E.: Semi-supervised learning: a brief review. Int. J. Eng. Technol. **7**(1.8), 81 (2018)
54. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning (Chapelle, o., et al., eds.; 2006) [book reviews]. IEEE Trans. Neural Netw. **20**(3), 542–542 (2009)
55. El Bouchefry, K., de Souza, R.S.: Learning in big data: Introduction to machine learning. In: Knowledge Discovery in Big Data from Astronomy and Earth Observation (pp. 225–249). Elsevier (2020)
56. Weerasekara, S., Li, W., Isaacs, J., Kamarthi, S.: Reinforcement learning for disassembly task control. Comput. Ind. Eng. **190**, 110044 (2024)
57. Lazaridis, P.C., Kavvadias, I.E., Demertzis, K., Iliadis, L., Vasiliadis, L.K.: Structural damage prediction of a reinforced concrete frame under single and multiple seismic events using machine learning algorithms. Appl. Sci. **12**(8), 3845 (2022)
58. Gu, G.X., Chen, C.T., Buehler, M.J.: De novo composite design based on a machine learning algorithm. Extreme Mech. Lett. **18**, 19–28 (2018)
59. Mangalathu, S., Jeon, J.S.: Classification of failure mode and prediction of shear strength for reinforced concrete beam-column joints using machine learning techniques. Eng. Struct. **160**, 85–94 (2018)
60. Young, B.A., Hall, A., Pilon, L., Gupta, P., Sant, G.: Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: new insights from statistical analysis and machine learning methods. Cem. Concr. Res. **115**, 379–388 (2019)
61. Feng, D.C., Liu, Z.T., Wang, X.D., Chen, Y., Chang, J.Q., Wei, D.F., Jiang, Z.M.: Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach. Constr. Build. Mater. **230**, 117000 (2020)
62. Dong, W., Huang, Y., Lehane, B., Ma, G.: XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring. Autom. Constr. **114**, 103155 (2020)
63. Pham, A.D., Ngo, N.T., Truong, T.T.H., Huynh, N.T., Truong, N.S.: Predicting energy consumption in multiple buildings using machine learning for improving energy efficiency and sustainability. J. Clean. Prod. **260**, 121082 (2020)
64. Ehteram, M., Ahmed, A.N., Sheikh Khozani, Z., El-Shafie, A.: Convolutional neural network-support vector machine model-gaussian process regression: a new machine model for predicting monthly and daily rainfall. Water Resour. Manage **37**(9), 3631–3655 (2023)
65. Hanoon, M.S., Ahmed, A.N., Razzaq, A., Oudah, A.Y., Alkhayyat, A., Huang, Y.F., El-Shafie, A.: Prediction of hydropower generation via machine learning algorithms at three Gorges Dam, China. Ain Shams Eng. J. **14**(4), 101919 (2023)
66. Karatas, I., Budak, A.: Development and comparative of a new meta-ensemble machine learning model in predicting construction labor productivity. Eng. Constr. Archit. Manag. **31**(3), 1123–1144 (2024)
67. Yi, Z., Luo, X.: Construction cost estimation model and dynamic management control analysis based on artificial intelligence. Iran. J. Sci. Technol. Trans. Civil Eng. **48**(1), 577–588 (2024)
68. Zhang, T., Xu, W., Wang, S., Du, D., Tang, J.: Seismic response prediction of a damped structure based on data-driven machine learning methods. Eng. Struct. **301**, 117264 (2024)
69. Zhang, F., Fleyeh, H., Wang, X., Lu, M.: Construction site accident analysis using text mining and natural language processing techniques. Autom. Constr. **99**, 238–248 (2019)
70. Kim, H., Ahn, E., Shin, M., Sim, S.H.: Crack and noncrack classification from concrete surface images using machine learning. Struct. Health Monit. **18**(3), 725–738 (2019)
71. Bhatta, S., Dang, J.: Machine learning-based classification for rapid seismic damage assessment of buildings at a regional scale. J. Earthquake Eng. 1–31 (2023)
72. Ravichandran, T., Gavahi, K., Ponnambalam, K., Burtea, V., Mousavi, S.J.: Ensemble-based machine learning approach for improved leak detection in water mains. J. Hydroinf. **23**(2), 307–323 (2021)
73. Fan, X., Zhang, X., Yu, X.B.: Machine learning model and strategy for fast and accurate detection of leaks in water supply network. J. Infrastruct. Preserv. Resilience **2**, 1–21 (2021)
74. Liu, C.C., Liu, J.: Damage identification of a long-span arch bridge based on support vector machine. Zhendong yu Chongji (J. Vibr. Shock) **29**(7), 174–178 (2010)

75. Figueiredo, E., Park, G., Farrar, C.R., Worden, K., Figueiras, J.: Machine learning algorithms for damage detection under operational and environmental variability. Struct. Health Monit. **10**(6), 559–572 (2011)
76. Liu, Z., Tesfamariam, S.: Prediction of lateral spread displacement: data-driven approaches. Bull. Earthq. Eng. **10**, 1431–1454 (2012)
77. Tezcan, J., Cheng, Q.: Support vector regression for estimating earthquake response spectra. Bull. Earthq. Eng. **10**, 1205–1219 (2012)
78. Marti-Vargas, J.R., Yepes, V., Ferri, F.J.: Prediction of the transfer length of prestressing strands with neural networks. Comput. Concrete Int. J. **12**(2), 169–186 (2013)
79. Naeej, M., Bali, M., Naeej, M.R., Amiri, J.V.: Prediction of lateral confinement coefficient in reinforced concrete columns using M5′ machine learning method. KSCE J. Civ. Eng. **17**, 1714–1719 (2013)
80. Park, J., Smarsly, K., Law, K.H., Hartmann, D.: Multivariate analysis and prediction of wind turbine response to varying wind field characteristics based on machine learning. Comput. Civil Eng. 113–120 (2013)
81. Jain, R.K., Smith, K.M., Culligan, P.J., Taylor, J.E.: Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Appl. Energy **123**, 168–178 (2014)
82. Jeon, J.S., Shafieezadeh, A., DesRoches, R.: Statistical models for shear strength of RC beam-column joints using machine-learning techniques. Earthquake Eng. Struct. Dynam. **43**(14), 2075–2095 (2014)
83. Butcher, J.B., Day, C.R., Austin, J.C., Haycock, P.W., Verstraeten, D., Schrauwen, B.: Defect detection in reinforced concrete using random neural architectures. Comput.-Aided Civil Infrastruct. Eng. **29**(3), 191–207 (2014)
84. Behnood, A., Verian, K.P., Gharehveran, M.M.: Evaluation of the splitting tensile strength in plain and steel fiber-reinforced concrete based on the compressive strength. Constr. Build. Mater. **98**, 519–529 (2015)
85. Geiß, C., Pelizari, P.A., Marconcini, M., Sengara, W., Edwards, M., Lakes, T., Taubenböck, H.: Estimation of seismic building structural types using multi-sensor remote sensing and machine learning techniques. ISPRS J. Photogramm. Remote Sens. **104**, 175–188 (2015)
86. Gonzalez, I., Karoumi, R.: BWIM aided damage detection in bridges using machine learning. J. Civ. Struct. Heal. Monit. **5**, 715–725 (2015)
87. Alimoradi, A., Beck, J.L.: Machine learning methods for earthquake ground motion analysis and simulation. J. Eng. Mech. **141**(4), 04014147 (2015)
88. Santos, A., Figueiredo, E., Silva, M.F.M., Sales, C.S., Costa, J.C.W.A.: Machine learning algorithms for damage detection: Kernel-based approaches. J. Sound Vib. **363**, 584–599 (2016)
89. Diez, A., Khoa, N.L.D., Makki Alamdari, M., Wang, Y., Chen, F., Runcie, P.: A clustering approach for structural health monitoring on bridges. J. Civ. Struct. Heal. Monit. **6**, 429–445 (2016)
90. Kaur, P., Dana, K.J., Romero, F.A., Gucunski, N.: Automated GPR rebar analysis for robotic bridge deck evaluation. IEEE Trans. Cybern. **46**(10), 2265–2276 (2016)
91. Vu, D.T., Hoang, N.D.: Punching shear capacity estimation of FRP-reinforced concrete slabs using a hybrid machine learning approach. Struct. Infrastruct. Eng. **12**(9), 1153–1161 (2016)
92. Rafiei, M.H., Adeli, H.: A novel machine learning-based algorithm to detect damage in high-rise building structures. Struct. Design Tall Spec. Build. **26**(18), e1400 (2017)
93. Gui, G., Pan, H., Lin, Z., Li, Y., Yuan, Z.: Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. KSCE J. Civ. Eng. **21**, 523–534 (2017)
94. Chatterjee, S., Sarkar, S., Hore, S., Dey, N., Ashour, A.S., Shi, F., Le, D.N.: Structural failure classification for reinforced concrete buildings using trained neural network-based multi-objective genetic algorithm. Struct. Eng. Mech. **63**(4), 429–438 (2017)
95. Karina, C.N., Chun, P.J., Okubo, K.: Tensile strength prediction of corroded steel plates by using machine learning approach. Steel Compos. Struct. **24**(5), 635–641 (2017)

96. Modarres, C., Astorga, N., Droguett, E.L., Meruane, V.: Convolutional neural networks for automated damage recognition and damage type identification. Struct. Control. Health Monit. **25**(10), e2230 (2018)
97. Zhang, Y., Burton, H.V., Sun, H., Shokrabadi, M.: A machine learning framework for assessing post-earthquake structural safety. Struct. Saf. **72**, 1–16 (2018)
98. Luo, H., Paal, S.G.: Machine learning–based backbone curve model of reinforced concrete columns subjected to cyclic loading reversals. J. Comput. Civ. Eng. **32**(5), 04018042 (2018)
99. Geyer, P., Singaravel, S.: Component-based machine learning for performance prediction in building design. Appl. Energy **228**, 1439–1453 (2018)
100. Kiani, J., Camp, C., Pezeshk, S.: On the application of machine learning techniques to derive seismic fragility curves. Comput. Struct. **218**, 108–122 (2019)
101. Mangalathu, S., Hwang, S.H., Choi, E., Jeon, J.S.: Rapid seismic damage evaluation of bridge portfolios using machine learning techniques. Eng. Struct. **201**, 109785 (2019)
102. Huang, H., Burton, H.V.: Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning. J. Build. Eng. **25**, 100767 (2019)
103. Yucel, M., Bekdaş, G., Nigdeli, S.M., Sevgen, S.: Estimation of optimum tuned mass damper parameters via machine learning. J. Build. Eng. **26**, 100847 (2019)
104. Zhu, J., Zhang, C., Qi, H., Lu, Z.: Vision-based defect detection for bridges using transfer learning and convolutional neural networks. Struct. Infrastruct. Eng. **16**(7), 1037–1049 (2020)
105. Mangalathu, S., Jang, H., Hwang, S.H., Jeon, J.S.: Data-driven machine learning-based seismic failure mode identification of reinforced concrete shear walls. Eng. Struct. **208**, 110331 (2020)
106. Prayogo, D., Cheng, M.Y., Wu, Y.W., Tran, D.H.: Combining machine learning models via adaptive ensemble weighting for prediction of shear capacity of reinforced-concrete deep beams. Eng. Comput. **36**(3), 1135–1153 (2020)
107. Chen, P.C., Chien, K.Y.: Machine learning based optimal seismic control of structure with active mass damper. Appl. Sci. **10**(15), 5342 (2020)
108. Kim, S.E., Vu, Q.V., Papazafeiropoulos, G., Kong, Z., Truong, V.H.: Comparison of machine learning algorithms for regression and classification of the ultimate load-carrying capacity of steel frames. Steel Compos. Struct. Int. J. **37**(2), 193–209 (2020)
109. Soize, C., Orcesi, A.: Machine learning for detecting structural changes from dynamic monitoring using probabilistic learning on manifolds. Struct. Infrastruct. Eng. **17**(10), 1418–1430 (2021)
110. Hwang, S.H., Mangalathu, S., Shin, J., Jeon, J.S.: Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete building frames. J. Build. Eng. **34**, 101905 (2021)
111. Kabir, M.A.B., Hasan, A.S., Billah, A.M.: Failure mode identification of column base plate connection using data-driven machine learning techniques. Eng. Struct. **240**, 112389 (2021)
112. Kim, B., Yuvaraj, N., Tse, K.T., Lee, D.E., Hu, G.: Pressure pattern recognition in buildings using an unsupervised machine learning algorithm. J. Wind Eng. Ind. Aerodyn. **214**, 104629 (2021)
113. Graciano, C., Kurtoglu, A.E., Casanova, E.: Machine learning approach for predicting the patch load resistance of slender austenitic stainless steel girders. In: Structures, vol. 30, pp. 198–205. Elsevier (2021)
114. Tong, Q., Couto, C., Gernay, T.: Machine learning models for predicting the resistance of axially loaded slender steel columns at elevated temperatures. Eng. Struct. **266**, 114620 (2022)
115. Tang, Q., Dang, J., Cui, Y., Wang, X., Jia, J.: Machine learning-based fast seismic risk assessment of building structures. J. Earthquake Eng. **26**(15), 8041–8062 (2022)
116. Bekdaş, G., Cakiroglu, C., Islam, K., Kim, S., Geem, Z.W.: Optimum design of cylindrical walls using ensemble learning methods. Appl. Sci. **12**(4), 2165 (2022)
117. Solhmirzaei, R., Salehi, H., Kodur, V.: Predicting flexural capacity of ultrahigh-performance concrete beams: machine learning–based approach. J. Struct. Eng. **148**(5), 04022031 (2022)
118. Chen, S.Z., Feng, D.C.: Multifidelity approach for data-driven prediction models of structural behaviors with limited data. Comput.-Aided Civil Infrastruct. Eng. **37**(12), 1566–1581 (2022)

119. Ocak, A., Nigdeli, S. M., Bekdaş, G., Işıkdağ, Ü.: Machine learning application of structural engineering problems. In: Hybrid Metaheuristics in Structural Engineering: Including Machine Learning Applications, pp. 179–198. Springer Nature Switzerland, Cham (2023)
120. Salmi, Z.J., Khodakarami, M.I., Behnamfar, F.: Development of seismic fragility curves for RC/MR frames using machine learning methods. Asian J. Civil Eng. **24**(3), 823–836 (2023)
121. Barkhordari, M.S., Tehranizadeh, M.: Data-driven dynamic-classifiers-based seismic failure mode detection of deep steel W-shape columns. Periodica Polytechnica Civil Eng. **67**(3), 936–944 (2023)
122. Naresh, M., Sikdar, S., Pal, J.: Vibration data-driven machine learning architecture for structural health monitoring of steel frame structures. Strain **59**(5), e12439 (2023)
123. Fan, C.L.: Evaluation of machine learning in recognizing images of reinforced concrete damage. Multimedia Tools Appl. **82**(19), 30221–30246 (2023)
124. Yu, S., Zhang, J.: Machine learning-based distraction-free method for measuring the optical displacement of long-span bridge structures. Struct. Control Health Monitor. 2024 (2024)
125. Cheng, J., Li, X., Jiang, K., Li, S., Su, A., Zhao, O.: Machine learning-assisted design of high strength steel I-section columns. Eng. Struct. **308**, 118018 (2024)
126. Ocak, A., Işıkdağ, Ü., Bekdaş, G., Nigdeli, S.M., Kim, S., Geem, Z.W.: Prediction of damping capacity demand in seismic base isolators via machine learning. CMES-Comput. Model. Eng. Sci. **138**(3) (2024)
127. Yan, Z., Zheng, S., Yang, F., Tai, X., Chen, Z.: A simplified approach to recognize vortex-induced vibration response using machine learning. Struct. Eng. Int. 1–13 (2024)
128. Yang, X.S.: Flower pollination algorithm for global optimization. In: Durand-Lose, J., Jonoska, N. (eds.) Lecture Notes in Computer Science, ed. 27, vol. 7445, pp. 240–249. Springer, London (2012)
129. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: an autocatalytic optimizing process. IEEE Trans. Syst. Man Cybern. B **26**, 29–41 (1996)
130. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature-Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74. Springer, Berlin, Heidelberg (2010)
131. Karaboğa, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization, vol. 200, pp. 1–10. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
132. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. **43**, 303–315 (2011)
133. Kazemzadeh Azad, S.: Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses. Eng. Optim. **50**(1), 89–105 (2018)
134. Jalili, S., Talatahari, S.: Optimum design of truss structures under frequency constraints using hybrid CSS-MBLS algorithm. KSCE J. Civ. Eng. **22**, 1840–1853 (2018)
135. Kooshkbaghi, M., Kaveh, A.: Sizing optimization of truss structures with continuous variables by artificial coronary circulation system algorithm. Iran. J. Sci. Technol. Trans. Civil Eng. **44**, 1–20 (2020)
136. Yousefpoor, H., Kaveh, A.: Chaos-embedded meta-heuristic algorithms for optimal design of truss structures. Scientia Iranica **29**(6), 2868–2885 (2022)
137. Altay, O., Cetindemir, O., Aydogdu, I.: Size optimization of planar truss systems using the modified salp swarm algorithm. Eng. Optim. 1–17 (2023)
138. Kaveh, A., Yousefpoor, H.: Chaotic optimization of trusses with frequency constraints with three metaheuristic algorithms. Iran. J. Sci. Technol. Trans. Civil Eng. **48**(1), 271–293 (2024)
139. Hieu, N.T., Cuong, N.Q., Tuan, V.A.: Optimization of steel roof trusses using machine learning-assisted differential evolution. J. Sci. Technol. Civil Eng. (JSTCE)-HUCE **15**(4), 99–110 (2021)
140. Nourian, N., El-Badry, M., Jamshidi, M.: Design optimization of truss structures using a graph neural network-based surrogate model. Algorithms **16**(8), 380 (2023)
141. Cao, H., Li, H., Sun, W., Xie, Y., Huang, B.: A boundary identification approach for the feasible space of structural optimization using a virtual sampling technique-based support vector machine. Comput. Struct. **287**, 107118 (2023)

142. Truong, V.H., Tangaramvong, S., Papazafeiropoulos, G.: An efficient LightGBM-based differential evolution method for nonlinear inelastic truss optimization. Expert Syst. Appl. **237**, 121530 (2024)
143. Pham, H.A., Dang, V.H., Vu, T.C., Nguyen, B.D.: An efficient k-NN-based rao optimization method for optimal discrete sizing of truss structures. Appl. Soft Comput. 111373 (2024)
144. Rao, R.: Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. **7**(1), 19–34 (2016)
145. The MathWorks, Matlab R2018a. Natick, MA (2018)
146. Python Software Foundation. PYTHON 3.11.0, USA (2022)
147. Sivasubramanian, A., Krishna, S.A., Nair, D.H., Varma, K., Radhakrishnan, R., Sathyan, D.: Experimental validation of compressive strength prediction using machine learning algorithm. Mater. Today: Proc. **64**, 181–187 (2022)
148. Ranstam, J., Cook, J.A.: LASSO regression. J. Br. Surg. **105**(10), 1348–1348 (2018)
149. Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970)
150. Rodrigues, A.P., Fernandes, R., Vijaya, P.: A study on the evaluation of different regressors in weather prediction. In 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE), pp. 13–18. IEEE (2022)
151. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B Stat Methodol. **67**(2), 301–320 (2005)
152. Ogutu, J.O., Schulz-Streeck, T., Piepho, H.P.: Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net, and their extensions. In: BMC Proceedings, vol. 6, pp. 1–6. BioMed Central (2012)
153. Vapnik, V., Guyon, I., Hastie, T.: Support vector machines. Mach. Learn. **20**(3), 273–297 (1995)
154. Kumar, S., Goyal, M.K., Deshpande, V., Agarwal, M.: Estimation of time dependent scour depth around circular bridge piers: application of ensemble machine learning methods. Ocean Eng. **270**, 113611 (2023)
155. Khan, M.S.I., Islam, N., Uddin, J., Islam, S., Nasir, M.K.: Water quality prediction and classification based on principal component regression and gradient boosting classifier approach. J. King Saud Univ.-Comput. Inform. Sci. **34**(8), 4773–4781 (2022)
156. Hashemizadeh, A., Maaref, A., Shateri, M., Larestani, A., Hemmati-Sarapardeh, A.: Experimental measurement and modeling of water-based drilling mud density using adaptive boosting decision tree, support vector machine, and K-nearest neighbors: a case study from the South Pars gas field. J. Petrol. Sci. Eng. **207**, 109132 (2021)
157. Rezaei, N., Jabbari, P.: K-nearest neighbors in R. Immunoinformatics of Cancers: Practical Machine Learning Approaches Using R, pp. 181–190. Academic Press (2022)
158. Zhang, Z.: Introduction to machine learning: k-nearest neighbors. Ann. Transl. Med. **4**(11) (2016)
159. Zhang, Z.: Too many covariates in a multivariable model may cause the problem of overfitting. J. Thoracic Dis. **6**(9) (2014)
160. Geisser, S., Eddy, W.F.: A predictive approach to model selection. J. Am. Stat. Assoc. **74**(365), 153–160 (1979)
161. Zhang, W., Wu, C., Li, Y., Wang, L., Samui, P.: Assessment of pile drivability using random forest regression and multivariate adaptive regression splines. Georisk: Assess. Manage. Risk Eng. Syst. Geohazards **15**(1), 27–40 (2021)

A Multi-objective Optimal Design Process for Determination of Link Capacity Expansions

İlyas Cihan Aksoy and Serdar Çarbaş

Abstract In recent years, especially in city centres, the rapid increase in the number of vehicles and mobility has triggered transportation-based problems such as traffic congestion, air pollution, and traffic accidents. In order to mitigate these emerging problems, the expansion of link capacities, which is a strategic decision falling under the class of the Road Network Design Problem, seems like a prominent solution method. Although expanding link capacities is an efficient way in terms of reducing travel times of network users, on the other hand, causing the infrastructure investment costs which is needed for expanding link capacities. Hence, within the scope of this study, the principal objective is to identify the optimal capacity increase of links for simultaneously minimizing the total travel times for all users and the infrastructure investment cost of the operator. Addressing this non-convex and NP-Hard design problem entails the utilization of a metaheuristic which is capable of reaching near or near-optimal design solutions for such type of complex engineering problems. To that end, a bi-level optimization model whose upper level generates candidate solutions through the NSGA-III along iterations to reach optimal solutions is proposed. The lower level of the bi-level optimization model calculates, for each candidate solution generated in the upper level, the total travel cost for all users by employing the static user equilibrium traffic assignment model based on Frank-Wolfe algorithm and the infrastructure investment cost derived from expanding link capacities. Numerical experiments are conducted on Sioux Falls Network which has been frequently utilized in prior studies concerning the problem. To better show the impact of different values for the parameters of the NSGA-III on the solution quality, a parameter tunning process is also undertaken. The Pareto optimal solutions of the best Pareto front obtained are compared with those previously reported studies. Additionally, a scenario analysis concerning the monetary (economical) costs for unit link capacity expansions is conducted, as well.

İ. C. Aksoy (✉) · S. Çarbaş
Department of Civil Engineering, Karamanoglu Mehmetbey University – Yunus Emre Campus, 70200 Karaman, Türkiye
e-mail: icihanaksoy@kmu.edu.tr

S. Çarbaş
e-mail: scarbas@kmu.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_4

Keywords Road network design problem · Link capacity expansion · Multi-objective optimization · Bi-level optimization

1 Introduction

In the last few decades, the rapidly increasing number of vehicles and human mobility has led to traffic congestion and consequently an increase in travel times, particularly in urban areas. The adverse effects of traffic congestion not only cause to transportation problems but also threaten public health, such that air pollutants considerably impact on the respiratory health. Furthermore, longer travel times cause a substantial increase in greenhouse gas emissions due to excessive fuel consumption, thereby triggering global warming and climate change [1]. On the other hand, motor vehicles constitute the primary source of noise [2], and scientific findings indicate that exposure to noise threatens human health by leading to health problems such as hypertension, hearing impairments, and ischemic heart disease [3]. It is needless to emphasize that traffic congestion leads to an increase in accidents. To mitigate emerging transportation-based problems and to create sustainable and eco-friendly cities and urban transportation, expanding road capacities is one of the first solution ways that firstly come to the minds of transportation planners and decision-makers. In this regard, [4] emphasizes that expanding road capacity contributes to an increase in demand for transportation, returning traffic congestion. However, [5] expresses that this solution method might be inescapable to alleviate traffic congestion.

The link capacity expansion problem is a design optimization problem in which capacity increases for a given set of links in a network are determined under multi constraints and falls into the class of the Urban Transportation Network Design Problem (UTNDP) which spans the planning, design, and management issues related to transportation. [6] divides this extensive problem into two sub-problems as Road Network Design Problem (RNDP) and Public Transportation Network Design Problem (PTNDP) because of involving not only private transport but also public transport. Furthermore, it also separates the RNDP into three subcategories based on the structure of decision variables, indicating that the problem of expanding road capacities falls within the class of Continuous Network Design Problem (CNDP). [7] expresses that the UTNDP handles a decision-making process that consists of strategic, tactical, and operational decisions in the field of transportation planning, in addition, inclining that strategic decisions are long-terms and associated with the transportation network infrastructure. In accordance with this definition, it becomes evident that the problem at hand is one of the strategic decisions of transportation planning.

Network design problems in transportation differs slightly from the design problems in other disciplines because it is necessary to take into account the potential reaction of the users to the changes in network. UTNDPs include two important cases; (i) the first one is to make decisions for the development of the network, (ii)

the second one is to estimate users' reaction opposite these decisions since the decisions taken by the leader or the decision maker, are effective on the users and their transportation pattern. Therefore, transportation network design problems should be considered as a bi-level problem. The upper-level problem is associated with the benefit of the decision maker, and contains measurable targets and constraints. The upper-level problem assumes that the decision maker is able to predict the behavior of the users. The lower-level problem is the problem of the users, and involves determining whether the users will travel or not and the path they will travel [6].

The link capacity expansion problem, like all UTNDPs, is inherently combinatorial and a multi-constrained design optimization problem characterized a vast search space where evaluation of all candidate solutions is time-consuming. Hence, the nature of the problem is NP-Hard, and addressing the problem through exact solution methods becomes a useless approach [8]. This means that it's extremely difficult to reach an optimal solution or near-optimal solution [7]. What is more, the non-convexity of the problem rises its complexity [9]. For these reasons, metaheuristic algorithms emerge as capable and convenient tools for solving such optimization problems in an acceptable duration.

When designing transportation networks, it is necessary to take two primary cost components into account: the total user cost for all users and the cost of decision maker responsible for designing the network. In transportation networks, the users seek to minimize their travel costs while reaching their destinations. To facilitate this desire of users, governments or decision makers expand link capacities to mitigate the congestion on roads and to shorten their travel times. However, an expansion of link capacities necessitates an infrastructure investment cost, in which the construction cost required for capacity improvement of links, from the perspective of decision makers. On the other hand, the efforts of decision makers to reduce their infrastructure investment cost is likely to result in an increase in total travel cost. This inherent trade-off between users' satisfaction and decision maker's cost, well-known in UTNDPs, makes the problem at hand complicated and multi-objective. Therefore, the design problem should be addressed considering both conflicting cost components in order to provide more robust network designs [10], with the help of multi-objective metaheuristic algorithms.

This chapter addresses the link capacity expansion design problem with the aim of minimizing simultaneously the total travel time for all users and the required infrastructure investment cost, employing the Non-dominated Sorting Genetic Algorithm (NSGA-III), a relatively novel multi-objective metaheuristic in the discipline of the UTNDPs. The developed bi-level optimization model was numerically tested on a specific subset of links on the Sioux Falls Networks. In order to better demonstration of the effect of different parameter values of the NSGA-III on the solution quality and to determine the best values for its parameters, a parameter tunning process was also conducted. The Pareto optimal solutions of the best Pareto front obtained were compared to the best solutions of the previously reported studies. Additionally, to demonstrate the impact of the assumed monetary (economical) cost per unit link capacity expansion on the results, a scenario analysis was executed considering different values for the monetary costs.

This chapter is structured in six sub-sections as follows. The next Sect. 2 presents a literature review, revealing the gap of knowledge and highlighting the scientific contribution of this chapter. Section 3 describes the design problem step-by-step manner, while Sect. 4 introduces the bi-level optimization model based on the NSGA-III. Subsequently, the computational results are presented in Sect. 5. Finally, Sect. 6 provides conclusions and suggestions for future researches.

2 Literature Review, Gap of Knowledge, and Contribution

In the past two decades, a substantial body of researches has been carried out for link capacity expansion problem by adopting varied objectives, decision variables, and solution approaches. [11], one of the early studies for the handled design problem, determined the optimal link capacity expansions using the Simulated Annealing for three different objectives; (i) minimizing the total travel cost, (ii) maximizing the reserve capacity, and (iii) the weighted sum of both, depending on a budget constraint. The developed model was applied to two small-size networks. [12] proposed an optimization model based on the Artificial Bee Colony that considers the minimization of sum of total travel time and weighted investment cost, testing on a toy network. The works of [13–17] and [18] addressed the problem with the aim of minimizing sum of total travel time and weighted investment cost, employing the augmented Lagrangian method, the four variants of Gradient-based Algorithms, the Conjugate Subgradient Projection Method, the Genetic Algorithm, the Cuckoo Search Algorithm, and the Harmony Search Algorithm, respectively. All the proposed optimization models were implemented on the Sioux Falls Network, a well-known benchmark network. As for the study of [19], a bi-level optimization model, which minimizes the total system travel time under a budget constraint, was developed for the problem by using a relaxation algorithm for solving the problem. Lastly, [20] employed a heuristic algorithm to determine the contributive degree of the link expansion effect on network capacity.

When examining the aforementioned literature studies, it can be seen that the intricate objectives were transformed into a single objective by applying the weighted sum method or a conversion factor to the investment cost. However, the selection of weights or conversion coefficients is relative, and changes in these values, that is, a change in the importance of objective functions bring into the open different solutions [21]. This emerges a considerable gap of knowledge. For this reason, to eliminate this issue, the recent studies in the other disciplines aim to obtain the set of Pareto optimal solutions. Actually, [22] considered to obtain the set of Pareto optimal solutions using the Chemical Reaction Optimization, still, ignoring the infrastructure investment cost operand, a vital absence for the problem. Hence, the quality of the solutions reached in this study is not comparable to the previous studies. Another gap of knowledge is that all the previously announced studies concerning the design problem consider a small subset of links in the network for expanding the link capacities, which distances the problem from being a real-world problem.

Overall, this chapter provides significant and unique contributions, listed below, to the existing body of the relevant literature considering the mentioned gaps of knowledge;

- This chapter address the link capacity expansion problem with the objective of attaining Pareto optimal solutions for the conflicting objectives of the problem, unlike the previous studies that rely on weights or conversion coefficients for objectives.
- This chapter utilizes the NSGA-III, a novel multi-objective metaheuristic, introducing it for the first time in the field of the UTNDPs to the best of our knowledge, thus aiming to provide valuable insights into the problem space.

3 Problem Description

This chapter focuses on determining the optimal capacity expansions for a given subset of links, whose capacities will be enhanced, on a road network. This decision-making process is undertaken by minimizing two distinct objectives; the total travel cost, Z_1, and the infrastructure investment cost, Z_2. . The former is the cost of traveling for all users in the network, whereas the latter is the cost derived from the construction of expanding road capacities. Thus, the total travel cost is temporal (in hours), while the infrastructure investment cost is monetary (in dollars).

Let us consider a directed road network denoted by $\mathbf{G} = (\mathbf{A}, \mathbf{N})$, where $\mathbf{N}$ is the set of nodes, $\mathbf{A}$ is the set of links in the road network. $\overline{\mathbf{A}}$ represents the set of links, capacities of which will be expanded, and $\mathbf{A} \subset \overline{\mathbf{A}}$. For each link a, $\forall a \in \mathbf{A}$, the corresponding link travel times, t_a, is determined using the strictly increasing link performance function developed by [23], as formulated in Eq. 1, where c_a and t_a^0 denote the capacity and free flow travel time of link a, respectively. $\tilde{x}_a$ is the flow (e.g., the number of vehicles) on link a, which satisfies the user-equilibrium criterion while $\hat{c}_a$ indicates the amount of the link capacity expansion if link a belongs to $\overline{\mathbf{A}}$. $\tilde{x}_a$ is derived through traffic assignment process where $\mathbf{Q}$ with the set of generic elements q_{rs} that represents the number of trips from origin node r, $r \in \mathbf{N}$ to node $s \in \mathbf{N}$, is appropriately assigned onto the road network. Ultimately, the total travel cost, Z_1, in the network, one of the objective functions is calculated using user-equilibrium flows, $\tilde{x}_a$, and associated with link travel times, t_a, as in Eq. 2.

$$t_a = t_a^0 \times \left(1 + 0.15 \times \left(\frac{\tilde{x}_a}{c_a + \hat{c}_a}\right)^4\right) \tag{1}$$

$$Z_1 = \sum_{a \in \mathbf{A}} \left(t_a\left(\tilde{x}_a, \hat{c}_a\right) \times \tilde{x}_a\right) \tag{2}$$

As for another objective function, the infrastructure investment cost, Z_2, is defined via Eq. 3 where d_a is the monetary cost (in dollars) per unit capacity expansion for link a. Actually, a different function structure could have been considered for the

infrastructure investment cost, such as a cost function that depends on the length of the link. However, it is important to note that all former studies regarding the problem employed this function. Therefore, for the sake of comparability with the existing literature, the present study adopts this formulation for the infrastructure investment cost.

$$Z_2 = \sum_{a \in \overline{\mathbf{A}}} \left(d_a \times \hat{c}_a^2 \right) \tag{3}$$

Overall, the mathematical program which gives the optimal capacity expansions, $\widetilde{\mathbf{c}}$, for links in $\overline{\mathbf{A}}$ is presented in Eq. 4. Additionally, an upper bound, b^u, for capacity expansions of these links is also considered, as expressed in Eq. 5.

$$\widetilde{\mathbf{c}} = \{\tilde{c}_1, \ldots, \tilde{c}_i \ldots, \tilde{c}_{nVar}\} = \left[\text{argmin } Z_1(\mathbf{G}, \mathbf{Q}, b^u), \text{argmin } Z_2(\overline{\mathbf{A}}, d_a, b^u) \right] \tag{4}$$

$$0 \leq \tilde{c}_i \leq b^u \quad \forall a \in \overline{\mathbf{A}} \tag{5}$$

4 Bi-level Optimization Model

The present chapter address the design problem of determining the optimum capacity expansions of links with the help of a bi-level optimization model. The upper level of the proposed bi-level framework utilizes the NSGA III, a multi-objective meta-heuristic, each step of which is detailed in the following subsection, to overcome this NP-Hard and combinatorial problem. The metaheuristic used generates a giant set of candidate solutions for the capacity expansions of the specified links, whose capacities will be expanded, along iterations. In order to reach the best solution considering the minimization targets of the total travel cost and the infrastructure investment cost, each of candidate solutions is evaluated in the lower level of the bi-level optimization model. The total travel cost related to a candidate solution for the link capacity expansion is accomplished through outputs obtained from the traffic assignment process. Figure 1 outlines an overall flowchart of the framework concerning the bi-level optimization model for solving the problem at hand.

4.1 Upper-Level Model

Early studies concerning the UTNDPs (including both RNDPs and PTNDPs) have addressed their problems considering a single objective, as reported in [6] and [24]. When it comes to the link capacity expansion problem, most of the former studies have focused on minimizing the weighted sum of intricate objectives, as can be seen

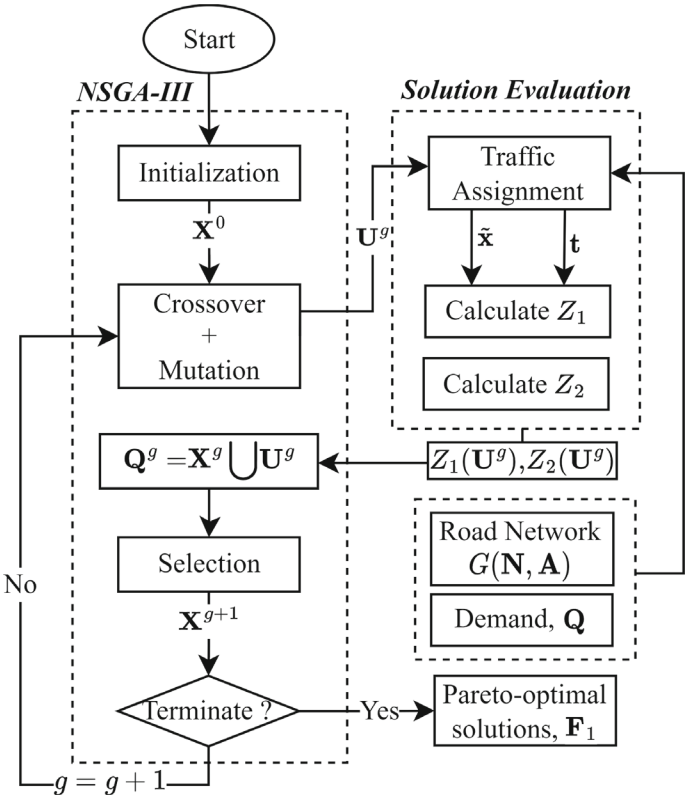


Fig. 1 Flowchart of the solution framework

in Sect. 2. [22], a recent investigation for the problem has adopted the approach of obtaining non-dominated solutions, a novel optimization insight for the UTNDPs. Motivated by the aforementioned facts, this study employes the NSGA-III, developed in the work of [25], for addressing the problem. The steps of the NSGA-III are explained extensively in the following paragraphs.

Let assume that the parent population with the size *nPop* at the generation g is denoted by $\mathbf{X}^g$. In the parent population $\mathbf{X}^g$, each chromosome or solution i is symbolized by $\mathbf{x}_i^g$, $i = \{1, \ldots, nPop\}$, thus $\mathbf{X}^g = \left\{\mathbf{x}_1^g, \ldots, \mathbf{x}_i^g, \ldots, \mathbf{x}_{nPop}^g\right\}$. Furthermore, each chromosome consists of a vector of *nVar* genes or variables, that is $\mathbf{x}_i^g = \left\{x_{i,1}^g, .., x_{i,j}^g, \ldots, x_{i,nVar}^g\right\}$, where *nVar* is the number of links in $\overline{\mathbf{A}}$. $x_{i,j}^g$ in $\mathbf{x}_i^g$ corresponds to the amount of link capacity expansion for j th link of $\overline{\mathbf{A}}$. In other words, $x_{i,j}^g$ is the amount of link capacity expansion, $\hat{c}_a$, for link a in $\overline{\mathbf{A}}$.

The NSGA-III starts with an initial population $\mathbf{X}^0$ whose each gene $x_{i,j}^0$ is chosen randomly, and aiming to direct the population toward the neighborhood of the global non-dominated solutions along with the generations. The algorithm terminates when

the maximum number of generations, nG, is reached. During this process, $\mathbf{x}_i^g$ is updated for the next generation in the following. Firstly, the offspring population, $\mathbf{U}^g$, is created from the parent population, $\mathbf{X}^g$, using the crossover and mutation operators which are associated with a crossover probability, p_c, and mutation probability, p_m. The arithmetical crossover and random mutation methods are used for the operators, respectively [26]. In the mutation operator, the genes to be mutated are selected randomly and the number of these genes is equal to$r_m \times nVar$. Then, the set of parent chromosomes, $\mathbf{X}^g$, and the set of offspring chromosomes, $\mathbf{U}^g$, are combined in the set$\mathbf{Q}^g$. In order to determine the subset of chromosomes in the set $\mathbf{Q}^g$ to be transferred to$\mathbf{X}^{g+1}$, the selection operator is employed.

The selection operator is briefly presented in the following steps: firstly, the Pareto fronts $\{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_l, \ldots\}$ is identified for the set $\mathbf{Q}^g$ by using the usual domination sorting process. Then, all solutions from the level 1 to the last level in terms of Pareto front domination are added one by one to the set $\mathbf{H}^g$. If $|\mathbf{H}^g| = nPop$ after any $\mathbf{F}_l$ is added to the set $\mathbf{H}^g$, the next generation $\mathbf{X}^{g+1}$ starts with the set $\mathbf{H}^g$ and no further consideration are needed: $\mathbf{X}^{g+1} = \mathbf{H}^g$. But, if $|\mathbf{H}^g| > nPop$, all solutions from the $\mathbf{F}_1$ to the $\mathbf{F}_{l-1}$ are already included; $\mathbf{X}^{g+1} = \bigcup_{j=1}^{l-1} \mathbf{F}_j$. The missing parent chromosomes in the set $\mathbf{X}^{g+1}$ are selected from the front $\mathbf{F}_l$ using the reference-point-based approach, which is the NSGA-III's own unique characteristic. This approach incorporates a sequence of sub-procedures: namely Normalization, Association and Niche-Preservation operations. The further details of the reference-point-based approach is presented step by step in the work of [25]. Overall, the brief procedure of the NSGA-III is given in Fig. 2.

4.2 Lower-Level Model

The quality of a candidate solution (i.e., the total travel cost and infrastructure investment cost of the solution) generated by the NSGA-III at the upper level is assessed at the lower level. The infrastructure investment cost, Z_2, is given by Eq. 3 where the calculation process is relatively easy. However, obtaining the total travel time, Z_1, is a challenging problem since the link flows and link travel times is achievable with traffic assignment models which are used to estimate the flows on a network by loading the demand for each OD pair onto the network. Therefore, the lower level of the bi-level optimization model employes the user equilibrium traffic assignment model, and allocates the demand according to the Wardrop's first principle which states that the travel times of all used paths between a specified OD pair are equal and less than those of the unused paths [27]. Such pattern of link flows can be found by solving the following mathematical program below developed by [28].

$$min \sum_{a \in \mathbf{A}} \int_0^{x_a} t_a\left(w, \hat{c}_a\right) dw \tag{6.a}$$

Input: $nPop, nVar, p_c, p_m, r_m, nZ, nG$
$\mathbf{X}^0 = Initialization(nPop, nVar)$
while $g \leq nG$
  $\mathbf{U}^g = \emptyset, \mathbf{H}^g = \emptyset$
  $\mathbf{U}^g = Crossover(\mathbf{X}^g, p_c) + Mutation(\mathbf{X}^g, p_m, r_m)$
  $\mathbf{Q}^g = \mathbf{X}^g \bigcup \mathbf{U}^g$
  $\{\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_l, ...\} = Sorting(\mathbf{Q}^g)$
  while $|\mathbf{H}^g| \geq nPop$
    $l = l + 1$
    $\mathbf{H}^g = \mathbf{H}^g \bigcup \mathbf{F}_l$
  end
  if $|\mathbf{H}^g| = nPop$
    $\mathbf{X}^{g+1} = \mathbf{H}^g$
  else
    $\mathbf{X}^{g+1} = \bigcup_{j=1}^{l-1} \mathbf{F}_j$
    $\mathbf{X}^{g+1} = \mathbf{X}^{g+1} \bigcup Normalize + Associate + Niche(\mathbf{F}_l, nZ)$
  end
  $g = g + 1$
end

Fig. 2 Brief procedure of the NSGA-III

$$\sum_{k \in \mathbf{K}} f_k^{rs} = q_{rs} \; \forall r \in \mathbf{R}, s \in \mathbf{S}, k \in \mathbf{K}_{rs} \tag{6.b}$$

$$x_a = \sum_{k \in \mathbf{K}} f_k^{rs} = q_{rs} \; \forall r \in \mathbf{R}, s \in \mathbf{S}, k \in \mathbf{K}_{rs} \tag{6.c}$$

$$f_k^{rs} \geq 0 \; \forall r \in \mathbf{R}, s \in \mathbf{S}, k \in \mathbf{K}_{rs} \tag{6.d}$$

The mathematical program which results in the link flows and link travel times on the user-equilibrium conditions is solved the Frank-Wolfe Algorithm that summarized as below [29].

Step 0: *Initialization.* Set the counter n to 1. Perform all-or-nothing assignment to the empty network based on $t_a(0), \forall a$. This yields $\mathbf{x}^n$, that is $\mathbf{x}^1$.

Step 1: *Update.* Calculate $t_a^n = t_a\left(x_a^n\right), \forall a$.

Step 2: *Direction finding.* Execute the all-or-nothing assignment based on $\mathbf{t}^n$. This yields a set of auxiliary flows, $\mathbf{y}^n$.

Step 3: *Line search.* Find α_n that solves $\min_{0 \leq \alpha \leq 1} \sum_{a \in \mathbf{A}} \int_0^{x_a^n + \alpha\left(y_a^n - x_a^n\right)} t_a(w) dw$

Step 4: *Move.* Calculate $x_a^{n+1} = x_a^n + \alpha_n\left(y_a^n - x_a^n\right), \forall a$.

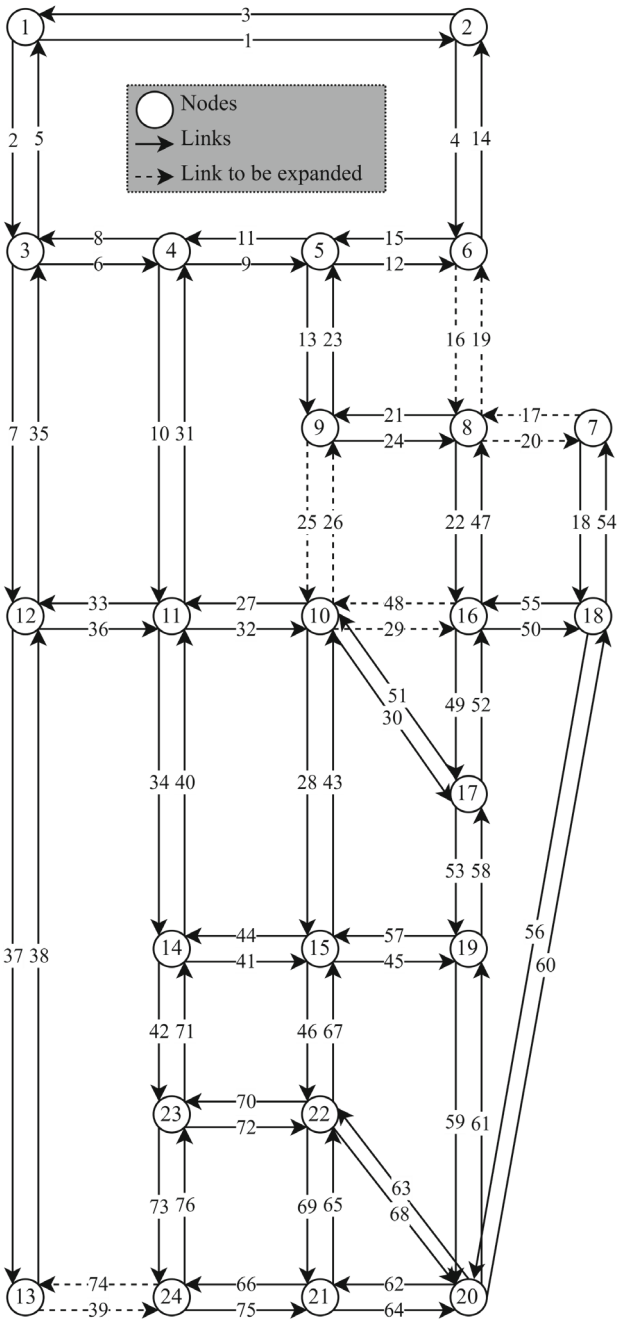
Step 5: *Convergence test*. If $\frac{\sqrt{\sum_{a\in\mathbf{A}}\left(x_a^{n+1}-x_a^n\right)^2}}{\sum_{a\in\mathbf{A}}x_a^n} \leq \kappa$, set $\mathbf{x}^{n+1}$ as $\tilde{\mathbf{x}}$. Otherwise, $n = n+1$ and go to Step 1.

5 Numerical Experiments

The developed optimization model was applied to Sioux Falls Network, which has been frequently used in previous studies and consists of 24 nodes and 76 links. The relevant network and associated link attributes (free-flow travel time and capacity) are given in Fig. 3. Besides, in Fig. 3, links 16, 17, 19, 20, 25, 26, 29, 39, 48 and 74 that are candidates for capacity expansion are depicted by the dashed line and their cost per unit capacity expansion are also given. In addition, the demand matrix for all OD pairs for this road network for the peak hour is given in Table 2. It should be noted that, the demand matrix and the values used for link attributes are the same as the previous studies for the comparative purpose. The proposed optimization model is coded on MATLAB 2021b, and conducted on an Intel Core i7 computer with a 3.6 GHz CPU of 16 GB RAM. The average processing time per optimization run of 100 generations (*nG*) with 40 populations (*nPop*) takes approximately two hours. The parameter values used in experiments are presented in Table 1.

5.1 Parameter Tunning Process

Metaheuristics, as capable algorithms for tackling such complex problems, are notably sensitive to the values of their parameters. Employing metaheuristics with the inconvenient values of parameters may lead to failure in attaining optimal or near-optimal solutions, particularly in problems with a large search space. Therefore, a parameter tuning process was conducted for the NSGA-III to better demonstrate the impact of different parameter values on the solution quality. The parameters *nPop*, *nG*, p_c, and p_m were not considered for the parameters tuning process since it is an expected fact that increasing these parameter values influence the quality of the solutions positively. At least, we are aware of that an increase in these values does not decrease the quality of the solutions. For this reason, their values were assumed as in Table 1. On the other hand, using different values for the parameter r_m may cause condiderable variations in the optimization results. To that end, to ascertain the optimal value of r_m specific the problem at hand, the brute-force approach where all candidate parameter values are executed by the same number on the problem [30] was adopted, and the candidate values for r_m were generated employing the full factorial design technique. Consequently, the NSGA-III was executed for four distinct values of r_m: 0.2, 0.4, 0.6, and 0.8. Furthermore, four optimization runs were carried out for each value ofr_m, resulting in 16 optimization runs in total.



Links	$c_a\left(\frac{v}{h}\right)$	$t_a^0(h)$	$d_a(\$)$
1-3	25.9002	0.06	
2-5	23.4035	0.04	
4-14	4.9582	0.05	
6-8	17.1105	0.04	
7-35	23.4035	0.04	
9-11	17.7828	0.02	
10-31	4.9088	0.06	
12-15	4.9480	0.04	
13-23	10.0000	0.05	
16-19	4.8986	0.02	26
17-20	7.8418	0.03	40
18-54	23.4035	0.02	
21-24	5.0502	0.10	
22-47	5.0458	0.05	
25-26	13.9158	0.03	25
27-32	10.0000	0.05	
28-43	13.5120	0.06	
29-48	5.1335	0.05	48
30-51	4.9935	0.08	
33-36	4.9088	0.06	
34-40	4.8765	0.04	
37-38	25.9002	0.03	
39-74	5.0913	0.04	34
41-44	5.1275	0.05	
42-71	4.9248	0.04	
45-57	15.6508	0.04	
46-67	10.3150	0.04	
49-52	5.2299	0.02	
50-55	19.6799	0.03	
53-58	4.8240	0.02	
56-60	23.4035	0.04	
59-61	5.0026	0.04	
62-64	5.0599	0.06	
63-68	5.0757	0.05	
65-69	5.2299	0.02	
66-75	4.8854	0.03	
70-72	5.0000	0.04	
73-76	5.0785	0.02	

Fig. 3 Sioux falls network

Table 1 Input values

Parameter	Value	Parameter	Value	Parameter	Value
nG	100	$nPop$	40	p_c	0.5
b^u	10	nZ	20	p_m	0.5
		κ	0.01		

In order to enable a comparative analysis among Pareto fronts resulted from different optimization runs, the hypervolume indicator which has been comprehensively explained in the work of [31] was employed. It is widely known that a higher hypervolume signifies a better Pareto front because it indicates that the Pareto front dominates a larger area of the objective space. The hypervolume reference point required for calculating hypervolume values of Pareto fronts is taken as the highest total travel time and the highest infrastructure investment cost among all Pareto fronts obtained. To identify the best value for r_m of the NSGA-III, the average hypervolume value for the four Pareto fronts generated by the relevant parameter value is used. As depicted in Fig. 4, which illustrates the average hypervolume value (I_H) for each value of r_m, it is evident that the proposed optimization model yields better Pareto fronts when executed by r_m of 0.8 with the average hypervolume value of 591,734. In other words, the solutions obtained by $r_m = 0.8$ are likely to exhibit more non-dominance. Briefly, it can be concluded that higher values for r_m may result in superior Pareto fronts.

5.2 *Best Pareto Front and Best Pareto Optimal Solution*

Figure 5 depicts the best Pareto front among 16 optimization experiments, exhibiting a hypervolume value of 595,510. This Pareto front consists of 40 Pareto optimal solutions, that is, containing as many solutions as $nPop$. As expected, this Pareto front was obtained by r_m of 0.8, which is determined as the best value for r_m through the parameter tunning process. Figure 6 displays the Pareto frontier concerning the best Pareto front, clearly demonstrating the intricate relationship between the total travel time and the infrastructure investment cost. To elucidate the differences between the solutions on the best Pareto front in terms of costs and the amount of capacity expansions, the detailed outputs for three Pareto optimal solutions highlighted by red, blue, and green colors in Fig. 6 are given in Table 3. It is noteworthy to note that the green solution exhibits the shortest Euclidean distance to the origin point (0,0) when considering the normalized costs of the solutions. Consequently, this green solution is treated as the best Pareto optimal solution along with the rest of this paper.

Table 3 depicts the comparative performance outputs among three solutions, demonstrating obviously the inherent trade-off between two objectives, such that the blue solution (the solution for the best infrastructure investment cost) effectively mitigates the infrastructure investment cost by limiting increases in link capacity

Table 2 Demand matrix

O/D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0.00	0.11	0.11	0.55	0.22	0.33	0.55	0.88	0.55	1.43	0.55	0.22	0.55	0.33	0.55	0.55	0.44	0.11	0.33	0.33	0.11	0.44	0.33	0.11
2	0.11	0.00	0.11	0.22	0.11	0.44	0.22	0.44	0.22	0.66	0.22	0.11	0.33	0.11	0.11	0.44	0.22	0.00	0.11	0.11	0.00	0.11	0.00	0.00
3	0.11	0.11	0.00	0.22	0.11	0.33	0.11	0.22	0.11	0.33	0.33	0.22	0.11	0.11	0.11	0.22	0.11	0.00	0.00	0.00	0.00	0.11	0.11	0.00
4	0.55	0.22	0.22	0.00	0.55	0.44	0.44	0.77	0.77	1.32	1.65	0.66	0.66	0.55	0.55	0.88	0.55	0.11	0.22	0.33	0.22	0.44	0.55	0.22
5	0.22	0.11	0.11	0.55	0.00	0.22	0.22	0.55	0.88	1.10	0.55	0.22	0.22	0.11	0.22	0.55	0.22	0.00	0.11	0.11	0.11	0.22	0.11	0.00
6	0.33	0.44	0.33	0.44	0.22	0.00	0.44	0.88	0.44	0.88	0.44	0.22	0.22	0.11	0.22	0.99	0.55	0.11	0.22	0.33	0.11	0.22	0.11	0.11
7	0.55	0.22	0.11	0.44	0.22	0.44	0.00	1.10	0.66	2.09	0.55	0.77	0.44	0.22	0.55	1.54	1.10	0.22	0.44	0.55	0.22	0.55	0.22	0.11
8	0.88	0.44	0.22	0.77	0.55	0.88	1.10	0.00	0.88	1.76	0.88	0.66	0.66	0.44	0.66	2.42	1.54	0.33	0.77	0.99	0.44	0.55	0.33	0.22
9	0.55	0.22	0.11	0.77	0.88	0.44	0.66	0.88	0.00	3.08	1.54	0.66	0.66	0.66	1.10	1.64	0.99	0.22	0.44	0.66	0.33	0.77	0.55	0.22
10	1.43	0.66	0.33	1.32	1.10	0.88	2.09	1.76	3.08	0.00	4.29	2.20	2.09	2.31	4.40	4.84	4.29	0.77	1.98	2.75	1.32	2.86	1.98	0.88
11	0.55	0.22	0.33	1.54	0.55	0.44	0.55	0.88	1.54	4.40	0.00	1.54	1.10	1.76	1.54	1.54	1.10	0.22	0.44	0.66	0.44	1.21	1.43	0.66
12	0.22	0.11	0.22	0.66	0.22	0.22	0.77	0.66	0.66	2.20	1.54	0.00	1.43	0.77	0.77	0.77	0.66	0.22	0.33	0.55	0.33	0.77	0.77	0.55
13	0.55	0.33	0.11	0.66	0.22	0.22	0.44	0.66	0.66	2.09	1.10	1.43	0.00	0.66	0.77	0.66	0.55	0.11	0.33	0.66	0.66	1.43	0.88	0.77
14	0.33	0.11	0.11	0.55	0.11	0.11	0.22	0.44	0.66	2.31	1.76	0.77	0.66	0.00	1.43	0.77	0.77	0.11	0.33	0.55	0.44	1.32	1.21	0.44
15	0.55	0.11	0.11	0.55	0.22	0.22	0.55	0.66	0.99	4.40	1.54	0.77	0.77	1.43	0.00	1.32	1.65	0.22	0.88	1.21	0.88	2.86	1.10	0.44
16	0.55	0.44	0.22	0.88	0.55	0.99	1.54	2.42	1.54	4.84	1.54	0.77	0.66	0.77	1.32	0.00	3.08	0.55	1.43	1.76	0.66	1.32	0.55	0.33
17	0.44	0.22	0.11	0.55	0.22	0.55	1.10	1.54	0.99	4.29	1.10	0.66	0.55	0.77	1.65	3.08	0.00	0.66	1.87	1.87	0.66	1.87	0.66	0.33
18	0.11	0.00	0.00	0.11	0.00	0.11	0.22	0.33	0.22	0.77	0.11	0.22	0.11	0.11	0.22	0.55	0.66	0.00	0.33	0.44	0.11	0.33	0.11	0.00
19	0.33	0.11	0.00	0.22	0.11	0.22	0.44	0.77	0.44	1.98	0.44	0.33	0.33	0.33	0.88	1.43	1.87	0.33	0.00	1.32	0.44	1.32	0.33	0.11
20	0.33	0.11	0.00	0.33	0.11	0.33	0.55	0.99	0.66	2.75	0.66	0.44	0.66	0.55	1.21	1.76	1.87	0.44	1.32	0.00	1.32	2.64	0.77	0.44
21	0.11	0.00	0.00	0.22	0.11	0.11	0.22	0.44	0.33	1.32	0.44	0.33	0.66	0.44	0.88	0.66	0.66	0.11	0.44	1.32	0.00	1.98	0.77	0.55
22	0.44	0.11	0.11	0.44	0.22	0.22	0.55	0.55	0.77	2.86	1.21	0.77	1.43	1.32	2.86	1.32	1.87	0.33	1.32	2.64	1.98	0.00	2.31	1.21
23	0.33	0.00	0.11	0.55	0.11	0.11	0.22	0.33	0.55	1.98	1.43	0.77	0.88	1.21	1.10	0.55	0.66	0.11	0.33	0.77	0.77	2.31	0.00	0.77
24	0.11	0.00	0.00	0.22	0.00	0.11	0.11	0.22	0.22	0.88	0.66	0.55	0.88	0.44	0.44	0.33	0.33	0.00	0.11	0.44	0.55	1.21	0.77	0.00

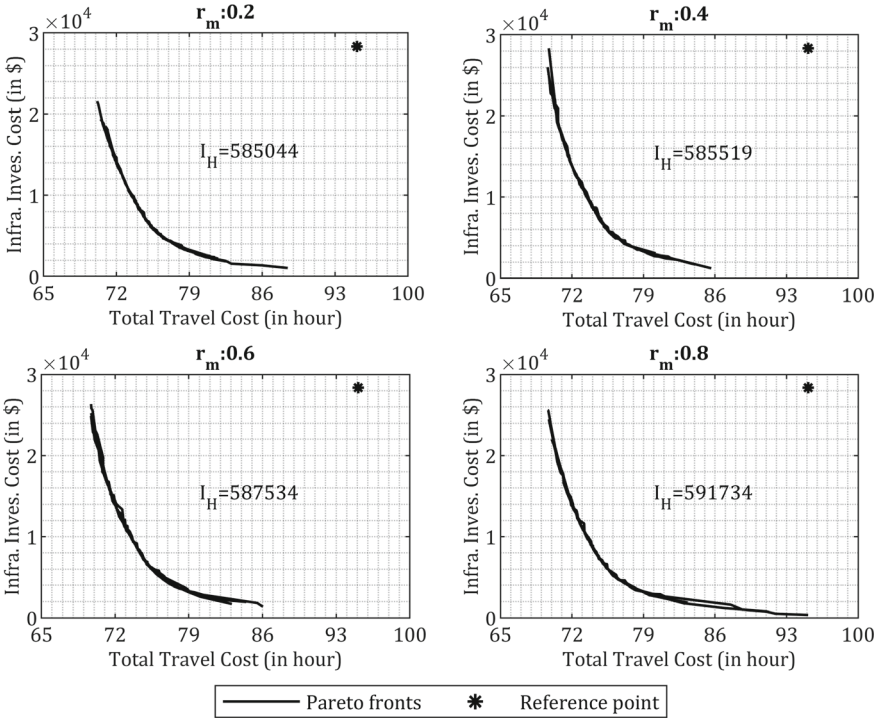


Fig. 4 Average hypervolumes for four values of r_m

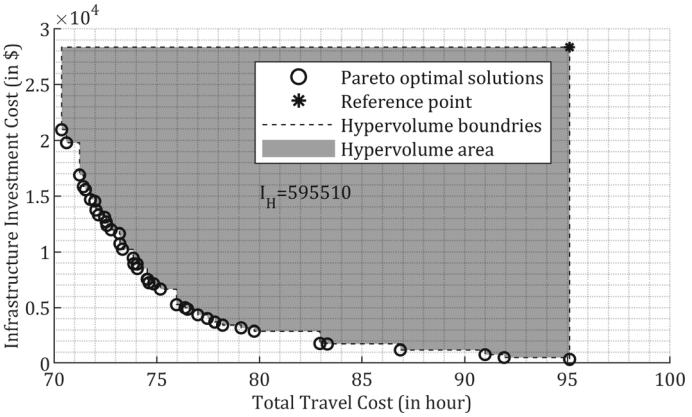


Fig. 5 Hypervolume of the best Pareto optimal solution

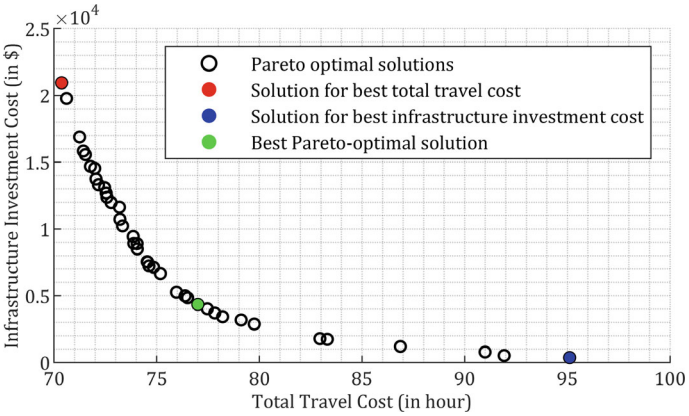


Fig. 6 Pareto frontier of the best Pareto optimal solution

Table 3 Comparative outputs of three solutions

	Red	Green	Blue
$\tilde{c}_{16}$	7.73	4.05	0.00
$\tilde{c}_{17}$	3.89	2.80	2.20
$\tilde{c}_{19}$	6.79	5.00	1.34
$\tilde{c}_{20}$	3.33	1.95	0.20
$\tilde{c}_{25}$	9.75	2.13	1.44
$\tilde{c}_{26}$	8.50	2.63	0.00
$\tilde{c}_{29}$	8.97	3.34	0.53
$\tilde{c}_{39}$	8.56	4.67	1.24
$\tilde{c}_{48}$	9.72	3.41	0.17
$\tilde{c}_{74}$	7.80	4.49	0.00
$Z_1(hour)$	70.37	77.00	95.11
$Z_2(\$)$	20,944	4,348	361

expansions, or vice versa in the red solution where the total travel cost is the highest in this front. Also, there exists a substantial difference in terms of both costs, especially the infrastructure investment cost. In terms of total travel cost, the blue solution is 1.35 times that of the red solution, while in terms of infrastructure investment cost, the red solution is approximately 58 times that of the blue solution. Furthermore, it is important to note that, link capacity expansions for mutual links connecting same nodes exhibit similar patterns. This is because, bi-directional links have the same performance characteristics, and the demand is symmetric.

Figure 7 shows the convergence pattern of the best Pareto front over successive generations, revealing a notable deceleration in its evolutionary progression after the 50th generation. Figure 8 presents the evolution of the hypervolume of the best Pareto front along with generations, thus validating the *nG*'s value determined through a

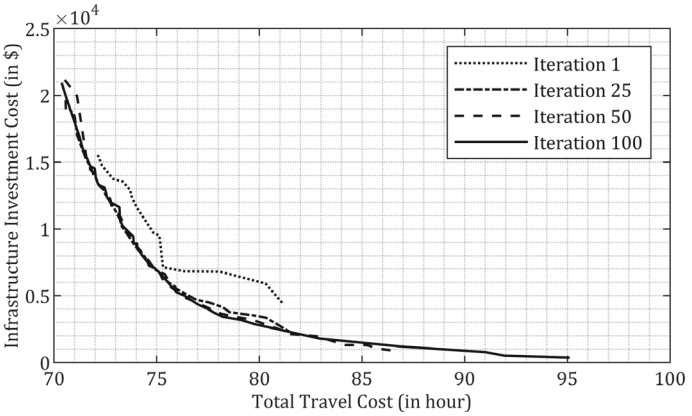


Fig. 7 Evolution of Best Pareto Front along with generations

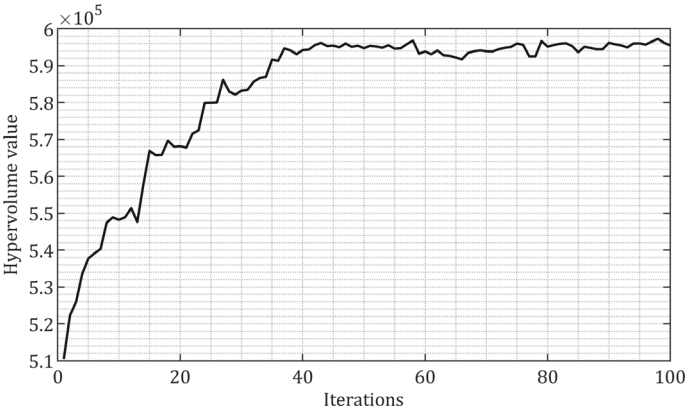


Fig. 8 Hypervolume for the Best Pareto Front along with generations

trial-and-error approach as 100 at the beginning of the computational experiments is an adequate generation number.

5.3 Comparative Results

To show the capability of the NSGA-III on the link capacity expansion design problem, the performance outputs obtained in this study are compared with the outputs observed in the previously published investigations.

For this comparison purpose, the best solutions obtained by the Augmented Lagrangian algorithm (AL) in [13], by the PARATAN version of gradient projection method (PT) in [14], by the Genetic Algorithm (GA) in [16], the Cuckoo Search (CS) algorithm in [17], and by the Harmony Search (HS) algorithm in [18] are used. The best solutions and the corresponding costs in these studies are presented in Table 4 where Z_1 and Z_2 are as obtained by applying our traffic assignment model to the resulted optimum link capacity expansions in the previous studies. As for Z^* and Z, the former is the weighted objective function value for the best solutions obtained in previous studies, the latter is the weighted sum of Z_1 and Z_2 through the structure of the weighted objective function for the best solutions of the previous studies. The reason why Z and Z^* are given in the table is to show that there may be a minor variation between them. This variation is derived from the operand of total travel cost since using different values for stopping criterions of traffic assignment algorithms employed in this study and previous studies leads to different sets of links flows, thus different total travel costs.

In Fig. 9, the best Pareto optimal solutions of the best Pareto front obtained and the best solutions of the previous studies are presented. When examining the figure, it can be seen that the solutions of AL and PT are clearly dominated by several Pareto optimal solutions while the solutions of GA, HS and CS are dominated by a Pareto optimal solution, which means that both Z_1 and Z_2 of these Pareto optimal solutions is smaller than those of the best solutions.

Table 4 Best solutions observed in previously published studies

	AL	PT	GA	CS	HS
$\tilde{c}_{16}$	5.5728	5.0237	5.17	5.0916	4.4482
$\tilde{c}_{17}$	1.6343	5.2158	2.94	1.3515	1.2926
$\tilde{c}_{19}$	5.6228	1.8298	4.72	6.4903	5.4675
$\tilde{c}_{20}$	1.6443	1.5747	1.76	2.2995	2.3064
$\tilde{c}_{25}$	3.1437	2.7947	2.39	2.9074	0.6453
$\tilde{c}_{26}$	3.2837	2.6639	2.91	2.0515	2.7100
$\tilde{c}_{29}$	7.6519	6.1879	2.92	3.6725	4.1596
$\tilde{c}_{39}$	3.8035	4.9624	5.99	5.2202	3.6761
$\tilde{c}_{48}$	7.3820	4.0674	3.63	3.4230	4.9047
$\tilde{c}_{74}$	3.6935	3.9199	4.43	4.8798	4.3878
$Z_1(hour)$	74.93	77.91	76.44	76.04	76.96
$Z_2(\$)$	8,743	6,320	5,027	5,316	4,864
Z	83.67	84.23	81.46	81.37	81.82
Z^*	81.75	82.53	81.74	81.51	81.83

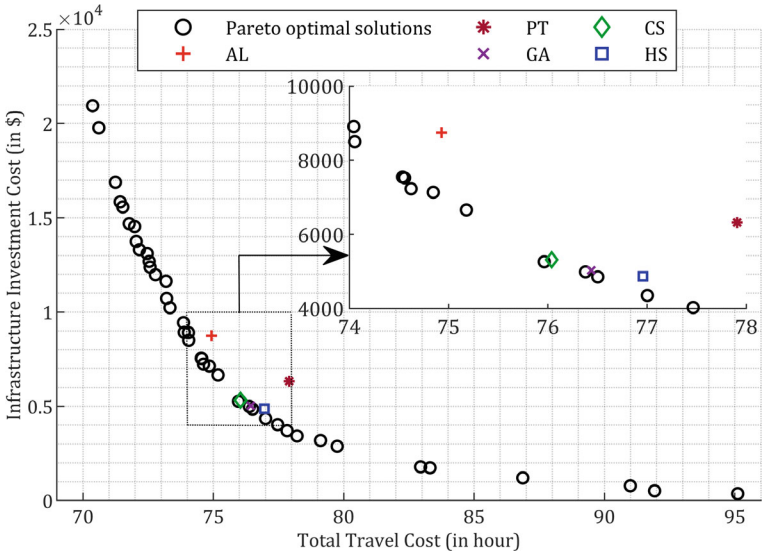


Fig. 9 Comparison between the best Pareto Front obtained and the best solutions of the previous studies

5.4 Scenario Analysis

This study assumes that the monetary costs per unit link capacity expansions are as listed in Fig. 3, consistent with all previous studies. However, it would be theoretically more realistic and logical to consider these monetary costs as values proportional to the length of the links (or the free-flow travel times of links). Therefore, in the alternative scenario given in Table 5, the alternative monetary costs adopted per unit link capacity expansion and the outputs of the optimization run executed with these alternative monetary costs are presented.

In this optimization run conducted for the alternative scenario analysis, the parameter values were assumed to be outlined in Table 1, and r_m was set to 0.8 which was

Table 5 Scenario analysis for monetary cost per unit capacity expansion

	Baseline Scenario	Alternative Scenario
$\{d_{16}, d_{17}, d_{19}, d_{20}, d_{25}, d_{26}, d_{29}, d_{39}, d_{48}, d_{74}\}$	{26,40,26,40,25, 25,48,34,48,34}	{20,30,20,30,30, 30,50,40,50,40}
Solution for best total travel cost (Z_1/Z_2)	70.37/20,944	69.88/23,593
Best Pareto optimal solution (Z_1/Z_2)	77.00/4,348	76.10/4,970
Solution for best inf. investment cost (Z_1/Z_2)	95.11/361	86.87/1,039
Hypervolume	595,510	597,034

determined as the best value through the parameter tunning process. The results indicate that when examining the best solutions for the infrastructure investment cost of the two scenarios, there are considerable differences in terms of both cost operands, emphasizing the importance of accurately determining monetary costs per unit link capacity expansion.

6 Conclusions

In this chapter, the link capacity expansion design problem, which falls into the class of the Road Network Design Problem, is addressed with the objective of concurrently minimizing the total travel cost for all users and the infrastructure investment cost. Given the NP-Hard and combinatorial nature of this problem, a bi-level optimization model whose upper level is based on the NSGA-III, a relatively novel multi-objective optimization method to the best of our knowledge for this field of the science, is proposed. The lower level of the bi-level model employs the Frank Wolfe Algorithm to determine link flows on user-equilibrium conditions. All numerical tests are executed on Sioux Falls Network which has been frequently utilized for the former studies related to the problem at hand. The parameter tunning process demonstrate that the NSGA-III can yield superior Pareto fronts when r_m is set to 0.8. It is obviously observed that the best Pareto optimal solutions on the resulting best Pareto front successfully dominate the best solutions achieved in previously published studies. Besides, the scenario analysis conducted reveals that monetary costs per unit link capacity expansion significantly determine the resulting costs.

This study can be extended in several directions for future researches. Firstly, this study assumes a uniform demand between origin–destination pairs for a given time period. However, in real-world transportation networks, demand varies along different periods of the day. Thus, utilization of dynamic traffic assignment models, which account for the fluctuation in demand over the reference period, will provide a more realistic approach in addressing the problem. Secondly, the performance of the NSGA-III on the design problem needs to be validated by comparing with other multi-objective metaheuristics. Lastly, it would be beneficial to consider a wider set of links whose capacities will be expanded, instead of a limited set of links, to better understand the performance of the proposed bi-level optimization model.

References

1. Bharadwaj, S., Ballare, S., Rohit, Chandel, M.K.: Impact of congestion on greenhouse gas emissions for road transport in Mumbai metropolitan region. Transp. Res. Procedia **25**, 3538–3551 (2017).https://doi.org/10.1016/j.trpro.2017.05.282
2. Delucchi, M.A., Hsu, S.L.: The External Damage Cost of Direct Noise from Motor Vehicles, Report No. UCD-ITS-RR-96–3 (4), Institution of Transportation Studies, University of California,Davis (1996)

3. Passchier-Vermeer, W., Passchier, W.F.: Noise exposure and public health. Environ. Health Perspect. **108**, 123–131 (2000). https://doi.org/10.1289/ehp.00108s1123
4. Tennøy, A., Tønnesen, A., Gundersen, F.: Effects of urban road capacity expansion—Experiences from two Norwegian cases. Transp. Res. Part D: Transp. Environ. **69**, 90–106 (2019). https://doi.org/10.1016/j.trd.2019.01.024
5. Anupriya, Bansal, P., Graham, D.J.: Congestion in cities: can road capacity expansions provide a solution? Transp. Res. Part A: Policy Pract. **174**, 103726 (2023). https://doi.org/10.1016/j.tra.2023.103726
6. Farahani, R.Z., Miandoabchi, E., Szeto, W.Y., Rashidi, H.: A review of urban transportation network design problems. Eur. J. Oper. Res. **229**, 281–302 (2013). https://doi.org/10.1016/j.ejor.2013.01.001
7. Magnanti, T.L., Wong, R.T.: Network design and transportation planning: models and algorithms. Transp. Sci. **18**(1), 1–55 (1984). https://doi.org/10.1287/trsc.18.1.1
8. Ben-Ayed, O., Boyce, D.E., Blair, C.E.: A general bilevel linear programming formulation of the network design problem. Transp. Res. Part B **22B**(4), 311–318 (1988). https://doi.org/10.1016/0191-2615(88)90006-9
9. Luo, Z.Q., Pang, J.S., Ralph, D.: Mathematical programs with equilibrium constraints. Cambridge University Press (1996)
10. Miandoabchi, E., Farahani, R.Z.: Optimizing reserve capacity of urban road networks in a discrete Network Design Problem. Adv. Eng. Softw. **42**, 1041–1050 (2011). https://doi.org/10.1016/j.advengsoft.2011.07.005
11. Yang, H., Wang, J.Y.T.: Travel time minimization versus reserve capacity maximization in the network design problem. Transp. Res. Rec. **1783**(1), 17–26 (2002). https://doi.org/10.3141/1783-03
12. Baskan, O., Dell'Orco, M.: Artificial bee colony algorithm for continuous network design problem with link capacity expansions. 10th International Congress on Advances in Civil Engineering, Ankara, Türkiye, pp.17–19 (2012)
13. Meng, Q., Yang, H., Bell, M.G.H.: An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. Transp. Res. Part B: Methodol. **35**(1), 83–105 (2001). https://doi.org/10.1016/S0191-2615(00)00016-3
14. Chiou, S.W.: Bilevel programming for the continuous transport network design problem. Transp. Res. Part B Methodol. **39**, 361–383 (2005). https://doi.org/10.1016/j.trb.2004.05.001
15. Chiou, S.W.: A subgradient optimization model for continuous road network design problem. Appl. Math. Model. **33**(3), 1386–1396 (2009). https://doi.org/10.1016/j.apm.2008.01.020
16. Mathew, T.V., Sharma, S.: Capacity expansion problem for large urban transportation networks. J. Transp. Eng. **135**(7), 406–415 (2009). https://doi.org/10.1061/(ASCE)0733-947X(2009)135:7(406)
17. Baskan, O.: Determining optimal link capacity expansions in road networks using cuckoo search algorithm with Lévy flights. J. Appl. Math. **2013**, 718015 (2013). https://doi.org/10.1155/2013/718015
18. Baskan, O.: Harmony search algorithm for continuous network design problem with link capacity expansions. KSCE J. Civ. Eng. **18**, 273–283 (2014). https://doi.org/10.1007/s12205-013-0122-6
19. Wang, G., Gao, Z., Xu, M., Sun, H.: Joint link-based credit charging and road capacity improvement in continuous network design problem. Transp. Res. Part A: Policy Pract. **67**, 1–14 (2014). https://doi.org/10.1016/j.tra.2014.05.012
20. Yang, X.F., Liu, L.F.: Contributive degree evaluation of links for capacity expansion problem in urban traffic network. J. Comput. **30**, 31–44 (2019). https://doi.org/10.3966/199115992019083004003
21. Miettinen, K.: Introduction to Multiobjective Optimization: Interactive and Evolutionary Approaches. In: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (eds.) Multiobjective Optimization, pp. 1–26. Springer, Berlin Heidelberg, Berlin (2008)
22. Wang, Y., Szeto, W.Y.: Multiobjective environmentally sustainable road network design using pareto optimization. Comput. Aided Civil Infrastr. Eng. **32**(11), 964–987 (2017). https://doi.org/10.1111/mice.12305

23. U.S. Bureau of Public Roads: Traffic Assignment Manual, U.S. Department of Commerce, Washington DC, United States (1964)
24. Iliopoulou, C., Kepaptsoglou, K., Vlahogianni, E.: Metaheuristics for the transit route network design problem: a review and comparative analysis. Public Transport **11**(3), 487–521 (2019). https://doi.org/10.1007/s12469-019-00211-2
25. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: Solving problems with box constraints. IEEE Trans. Evolut. Comput. **18**(4), 577–601 (2014). https://doi.org/10.1109/TEVC.2013.2281535
26. Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. Artif. Intell. Rev. **12**(4), 265–319 (1998). https://doi.org/10.1023/A:1006504901164
27. Wardrop, J.G.: Some theoretical aspects of road traffic research. Proc. Inst. Civ. Eng. **1**(3), 325–362 (1952). https://doi.org/10.1680/ipeds.1952.11259
28. Beckmann, M., McGuire, C.B., Winsten, C.B.: Studies in the Economics of Transportation. Yale University Press (1955)
29. Sheffi, Y.: Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall Inc., New Jersey, USA (1985)
30. Huang, C., Li, Y., Yao, X.: A survey of automatic parameter tuning methods for metaheuristics. IEEE Trans. Evol. Comput. **24**(2), 201–216 (2020). https://doi.org/10.1109/TEVC.2019.2921598
31. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. Proceedings of the 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms, Orlando, Florida, pp. 87–102 (2009)

IoT with Deep Learning in Pipeline and Metro Track Risk Estimation Using Smart Cities Development

E. B. Priyanka, S. Thangavel, and Priyanka Prabhakaran

Abstract The Internet of Things (IoT) is a prolonged and improved gadget community primarily based totally at the Internet, and its closing purpose is to attain real-time interplay amongst things, machines and people through diverse superior technological platform of automation. Thus, the societal causes of oil pipeline risk analysis need continuous monitoring of vital signals from the physical to the cloud network. This is due to the fact IoT verbal exchange structures appreciably effect the software of sustainable clever towns, figuring out their scalability, stability, and computational output accuracy. Subsequently, to address those requesting circumstances of practical smart city plan, an incredible and dynamic verbal trade device is required; it should have the option to anticipating of risk analysis in the feature extraction. On contemplating versatile and viable verbal trade conditions essentially basically put together absolutely information transmission networks that bring down the attainable upgradation and power utilization towards the organization administration lifecycle. Hence, the deep learning application incorporated frameworks can perceive a smart town with extreme maintainable effectiveness on the oil pipeline network with proactive decision-making during maintenance cycle.

Keywords IoT · Deep learning · Smart cities

1 Introduction

The forecast of IoT contraption generally execution is mulled over of the urgent test because of the reality the IoT might be exceptionally confounded in nature on account of the enormous number of connected devices, huge length of traded information, unnecessary speed local area geography trade, and heterogeneous gadgets and

E. B. Priyanka (✉) · S. Thangavel · P. Prabhakaran
Department of Mechatronics Engineering, Kongu Engineering College, Perundurai, India
e-mail: priyankabhaskaran1993@gmail.com

P. Prabhakaran
Department of Civil Engineering, Kongu Engineering College, Erode, India

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_5

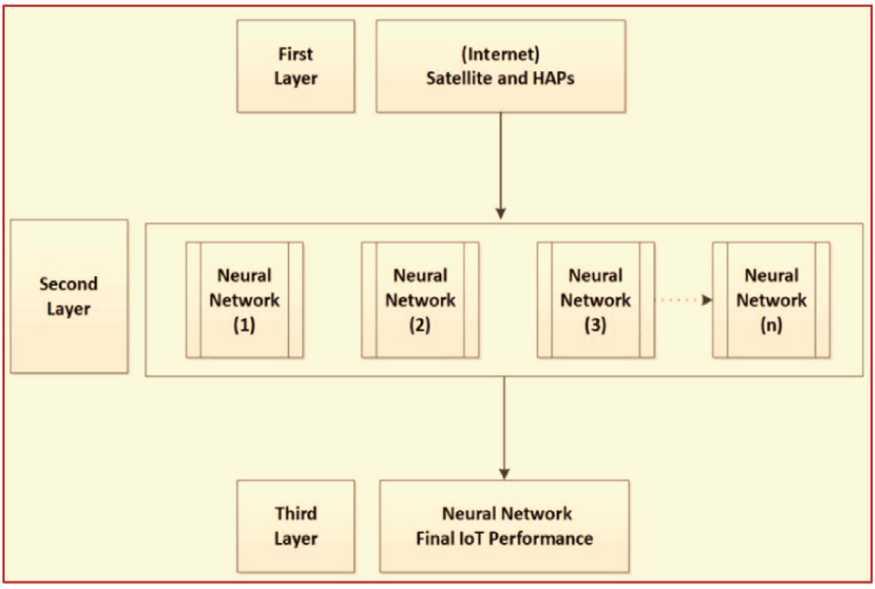


Fig. 1 Layered architecture for IoT with deep learning

associated things [1]. Moreover, IoT frameworks show non-hinder and fast changes in conduct. Along these lines, it sensibly speaking intense, if now not feasible, to perform IoT gadget standard generally execution forecast utilizing customary machine-becoming acquainted with models. Likewise, the total standard generally execution expectation for complete IoT frameworks, route to their unbalanced intricacy, stays to be an extreme assignment (Fig. 1).

Thusly, a DL model is proposed in this for foreseeing the overall generally speaking exhibition of an IoT gadget subsequent to changing it into a fixed of groups. Each layer contains at least one neural organization. The measure of neural organizations is resolved dependent on the measure of IoT organizations. These layers speak with each extraordinary utilizing their information sources and yields to expect the IoT gadget standard generally execution, it truly is the point of the DL model [2]. The proposed DL model uses the best importance of IoT (i.e., its foundation comprises of a large number including remote sensor organizations (WSNs), radio-recurrence recognizable proof (RFID) organizations, and mobileular impromptu organizations (MANETs) as given in Fig. 2. Edge computing plays background platform for decision making on the cloud server with respect to the application and physical architect layout and the reaction time of 1 carrier thru the collaboration of give up gadgets, area nodes, and the cloud. As proven in Fig. 2, the give up gadgets should calibrate the statistics computing challenge to the server and client nodes with minimum data transmission error rates and minimum payload with enhanced encryption scheme [3]. Taking the smart city application, video processing along with image classification supporting small level Artificial Intelligence keeps prediction rate at a higher

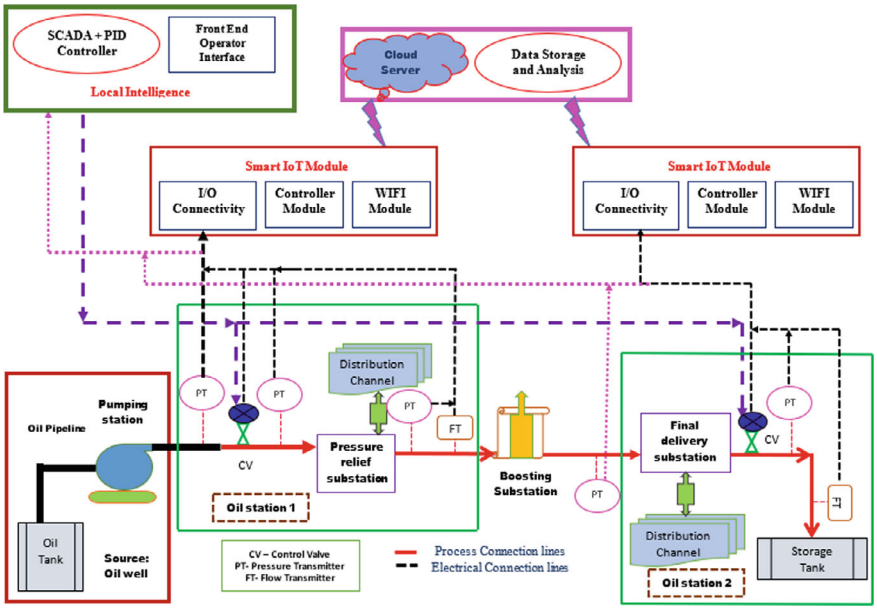


Fig. 2 Proposed Deep Learning incorporated test bed of oil pipeline experimental system

scale, accordingly keep away from the excessive latency and strength intake related to strolling a largescale AI version through them self.

However, IoT information commonly incorporates a tremendous quantity of person privateness, and person privateness might be discovered from that information. Edge nodes commonly do now no longer have the equal degree of safety as cloud centers, and processing the information on such area nodes would possibly cause an excessive danger of privateness leakage. Thus, a way to assure the safety of information and defend person privateness in collaborative area-cloud surroundings in destiny AI-enabled IoT is a difficulty that need to be addressed [4].

Perception of information involves creating and examining the visual representation of mathematical information. Persuasive representation is accessible to aid research and devise a collection of complex information to make it more available, reasonable and valuable. As a key parameter, data transfer of the preparation set initiates with handshaking level of higher security rate so that you can discover the relevance between some examples and negative examples. Each of the 3,967 Prep Set samples created are 102 measurement vectors, which means that each sample contains 102 highlights. In this study, two important nonlinear techniques, ISOMAP and Laplace Eigenmaps (LE), were used to recognize perception [5]. The prep set is clearly isolated into two classifications. On the contrary, it is assumed that the selection of elements in the preparation set is logical and that the information has a high level of perceptibility. A double arrangement is possible.

2 IoT Deep Learning Network Model Analysis

The framework of IoT may likewise besides exemplify WSN, RFID, MANET, and stand-out networks, which might be related withinside what's to come. IoT networks talk through the Web. In this manner, the overall by and large execution of the whole IoT device can be separated from the overall generally execution of each IoT segment. Accordingly, to are expecting the general presentation of a whole IoT apparatus, man or lady IoT network by and large execution esteems must be expected first. Henceforth, the proposed forecast model comprises of dynamic neural organizations. The overall generally execution of each neural organization is expected. At that point, the man or lady expected general by and large execution yields are utilized to are expecting the overall generally speaking exhibition of the whole IoT climate. Consequently, the assortment of neural organizations must be same to the assortment of IoT networks in addition to the neural organization that separates the last yield. The neural organization model comprises of 3 layers [6].

The primary layer is for the Web neural organization. This is a result of the truth the Web addresses the main transmission medium that speaks with unique organizations withinside the IoT instrument. The yield of the Web neural organization is utilized as a contribution for the second neural organization layer. This is a result of the truth the overall by and large execution of the Web immediately influences the exhibitions of different organizations; the verbal trade sign general by and large execution improvements must be considered withinside the expectation method of each organization withinside the IoT device. The 2nd layer of the proposed neural organization model incorporates various neural organizations same to the assortment of organizations withinside the IoT framework. Each neural organization predicts the overall generally execution of the IoT organization. The expected rate is viewed as a yield of the neural organization [7]. Thus, the assortment of yields withinside the 2d layer is same to the assortment of IoT networks less one (Web). These yields are viewed as contributions for the three layers of the neural organization model. The three layer incorporates the leftover neural organization, this is utilized to are expecting the last broad generally speaking presentation (by and large execution) of the whole IoT instrument as shown in Fig. 3.

Inferable from the utilization of IoT frameworks, which incorporates dynamic changes, adaptability, and colossal size, the expectation of the overall normal in general presentation stays to be an extreme undertaking. Thus, the IoT apparatus is part into various groups. Each bunch might be heterogeneous or homogenous. A heterogeneous group may furthermore in addition incorporate exact organizations which incorporates an assortment of hubs from WSN, RFID, and cell impromptu organizations. A homogenous bunch may also besides incorporate most straightforward one state of organization hub. The state of group (heterogeneous or homogenous), further to its size, is utilized to decide the assortment of covered up layers in each neural organization. Along these lines, our methodology can be portrayed as a unique neural organization approach. The normal generally execution forecast for an IoT neural organization essentially is predicated upon at the improvement

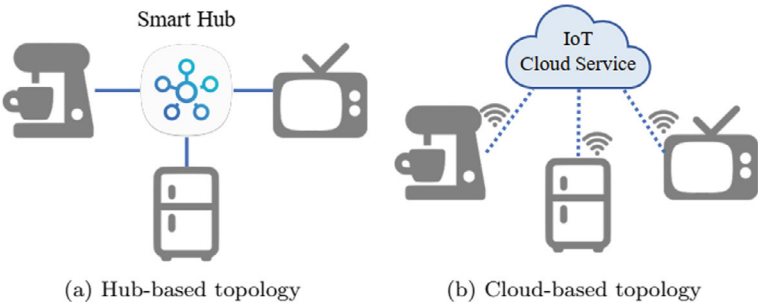


Fig. 3 Topology of IoT influence in data communication

of the Web neural organization. Thusly, the yield of the Web neural organization might be viewed as one of the contributions for every one of the option neural organizations. Thusly, each neural organization will create its yield. Subsequently, each neural organization yield might be viewed as a contribution to a couple of various neural organizations to separate the last yield. The last yield shows the forecast of the IoT instrument normal generally execution. Each neural organization conveys an extreme and fast of covered up layers comparatively to go into and yield layers. The center of each secret layer is the yield of its past secret layer.

3 Case Study Analysis on Implementation in Oil Pipeline Risk Prediction

The failure of Oil Pipelines is continually an unlucky incident due to the outcomes that it entails: in a few cases, the outcomes can be economical, environmental or, with-inside the worst conceivable condition, the injuries can initiate environment losses. The qualified excessive protection that the integrated machine has these days will not be been finished without the injuries that befell via antiquity. The reasons that brought about those injuries were, in a few cases, now no longer recognised earlier than they befell and, therefore, couldn't had been investigated previous to the accident. Sensors, faraway managed valves and precise software program can manipulate parameters, including running strain and others, which facilitate the manipulate of the machine and additionally enhance the leak detection and glide manipulate functionalities. Today, superior excessive-power steels including W232 and W891 are well evaluated for the capability utility of excessive strain and long-distance transportation pipelines, including the 8427 km pipe community in India where estimated running strain is 1998 psi. The medical network will not always blind to the implication of getting a green oil pipeline failure reporting machine. Although failure of city oil and fuel pipelines cannot be absolutely avoided, it could be decreased via way of means of green control and manipulate strategies.

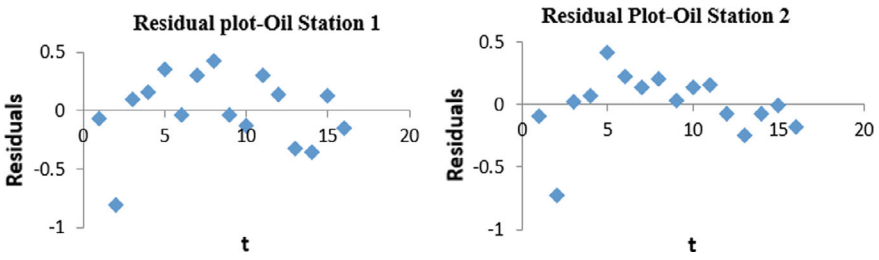


Fig. 4 Residual plot of oil station through risk estimation feature rates

In order to make sure dependable pipeline operations, machine integrity is vital to save you catastrophic screw ups and high-priced downtime. Pipeline machine integrity control is a "software that manages strategies, tools, and sports for assessing the fitness situations of pipelines and scheduling inspection and upkeep sports to lessen the dangers and costs". It is a technique which include 3 important steps: disorder detection and identification, disorder boom prediction, and risk-primarily based totally control. Significant advances are wanted in those 3 steps to correctly compare defects and pre-vent pipeline screw ups primarily based totally on inspection data, disorder boom prediction, and integrity sports optimization due to the fact the modern pipeline integrity strategies. In order to proceed with the implementation of risk analysis, initial residual analysis is taken to proceed to configure the layer fitting for deep neural network as shown in Fig. 4. This paper goals to growing a Failure Analysis Expert System supported via way of means of a set of rules via Digital dual mixed.

4 Prediction Analysis Based on Probability Assessment Using Digital Twin Platform

Digital Twin technology will help to upgrade the oil enterprise to integrate the work flow of sensors and actuators, as soon as a fault occurrence will always be detected and corrected, the oil transport pipeline overall performance might go to pot and in addition excessive protection troubles may want to occur. In order to offer automated safety selections for the duration of the operational degree withinside the context of business automation production, clever tracking and manage structures want to be evolved and deployed. Due to the upgradation of fourth business revolution, virtual twins are of important implication for business automation. Digital twins describe using holistic simulations to simply replicate a bodily system. Adopting virtual twins lets in operators to screen productions, check deviations in remoted digital surroundings, and in addition improve the safety of method industries. Extending from the paradigm layout of method factories, the cycle procedure includes nuclear virtual surroundings related to the actual world. Concerned with the progress of virtual twins,

the subspace identity method describes the complicated method via a state–area matrix, which presents an appealing digital replicate for the MIMO system.

Nevertheless, present SIMs are tough to assemble to be had digital fashions for complicated closed-loop structures. Therefore, an adaptive danger comparing structure wishes to be evolved to enhance the identity overall performance for virtual twins. The rationale of growing virtual twins is to enhance the safety of oil pipeline structures, and for that reason prediction of the automatic risk estimation of oil pipeline will be prevailing as the vital technologies for virtual twins. IoT based Automation layout permits method factories to preserve its predetermined operation after the pipeline injuries or danger takes place, in all likelihood at a discounted stage as opposed to a whole failure. Without thinking about the common-failure, a data-pushed primarily based totally method identity scheme is proposed to diagnose particular fault websites with the aid of using comparing the residual sign of the digital system. The new system primarily based totally on Fig. 5 designs residuals with preferred overall performance indices and arranges residual calculations similar to the manage strategy, and for that reason it's miles suitable for the automatic implementation towards the sensible method. As supplied in Fig. 6, the number one structure of the method identity scheme is to hit upon faults, discover fault types, and in addition diagnose particular fault websites.

To save engineers from fake alarms, the method identity is completed sequentially from the sensor physical layer to the output actuator nodes. Besides, the unchecked residual generator is constructed to keep away from generating indecisive consequences because of the oversensitivity. When an alarm takes place from the method tracking version on the time-span k, the scheme first computes the sensor residual parameter noted as rsen to discover the fault nature. If a sensor fault is identified, this system calculates the analysis orientated sensor residual rsen, dia to elect the desired fault feature. When the sensor residual rsen remains within the specified threshold, the actuator residual ract is computed to discover if the transportation

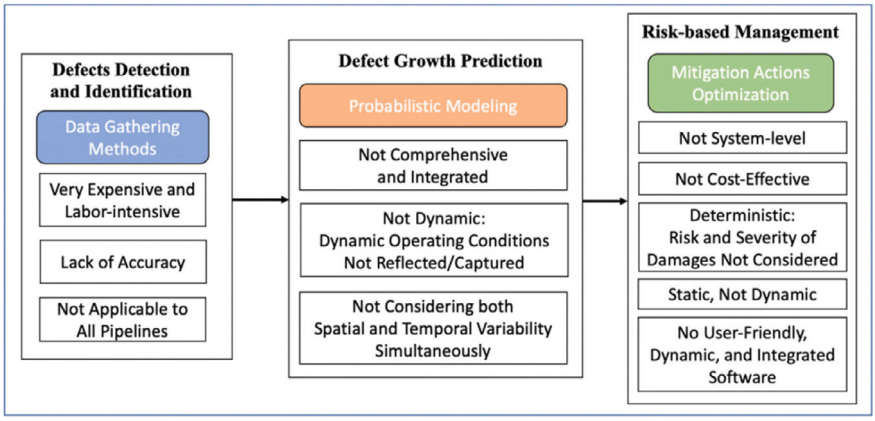


Fig. 5 Schematic description of the oil risk assessment approach in dynamic statistics

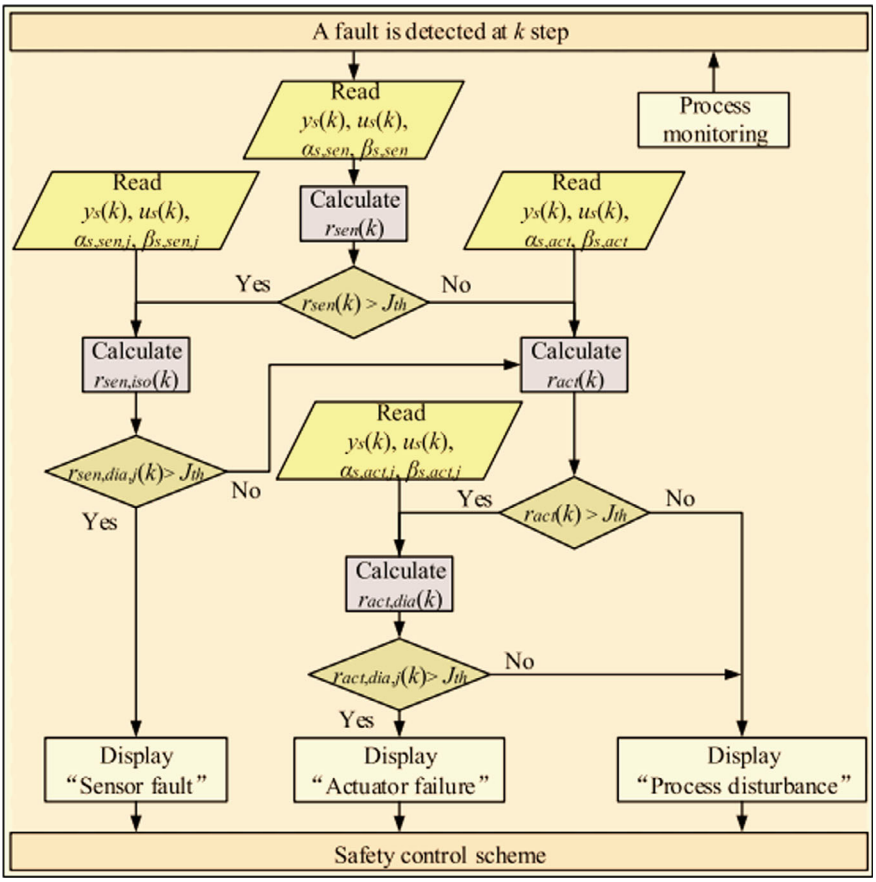


Fig. 6 Risk analysis on the feature extraction using digital twin

unit has an uncontrollable control valve actuation. Similar to the schematic work flow, the analysis orientated actuator residual ract, dia is calculated to decide the particular uncontrollable actuator while an actuator failure is detected, or the alarm is decided to be a fault of method disturbance. As said before, on the way to eliminate impulsive capabilities, predictability evaluation is completed on every check examination of multivariate time series risk analysis in oil pipelines. For better inference, simulation consequences from a random fault occurrence data ware house are undertaken and given for check are pronounced over distinct horizon indexes for higher understanding.

The acquired consequences display that capabilities F2, F3 do now no longer fulfil predictability standards neither with the aid of using DCP nor with the aid of using CC, as it's miles truly indicated with the aid of using their decrease values of predictability (P red approximately F2 and F3 are acquired from different check

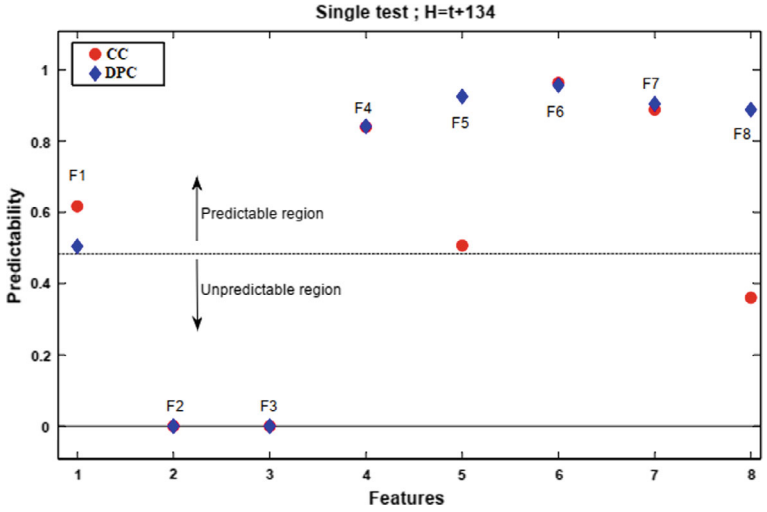


Fig. 7 Risk features predictability for the span of (H = t + 202)

instances as well, with the aid of using making use of each class tools. Moreover, DCP suggests higher overall presentation with better predictability standards in comparison to CC for maximum of the imitations.

This phenomenon is proven in Fig. 7, that depicts an assessment of the risk features probability rate of occurrences in the oil pipeline due to the enlargement of the network with minimum automation. The plot graph of time space with the minimum risk of 202 were presented with the notation of H = t + 202. The CC (Cumulative consideration) and DCP (Discrete Cumulative points) will be considered at the 8 zones with features processing (F) by grouping predictable and unpredictable regions. It is inferred that the 10:2 ratio scale, the features were taken and plotted for DCP and of Maximum Overlapping phenomenon of confirming the risk identification at the major rate of occurrences in the undertaken oil pipeline areas with earlier notification to the inspection engineer to reduce the societal losses.

5 Case Study on Routing Schedule on the Timing Protocols for the Metro Track Maintenance

Metro rail systems have been in service since 1984 providing ease of access to its commuters by reducing the passenger travel time and increase in commutation speed while increasing the service frequency. Gradually the metro rail systems have grown to its fullest across many cities in India. At present there are about thirteen operational metro rail systems with the largest network at Delhi covering a large span of 348.51 km. Most of the Indian metros are under the construction phase that accounts

to about 507.96 km, the entire span of operational network covers about 704.90 km, the approved routes sum up to 471.54 km, the proposed routes cover a whooping 1045 km. Transit system have been shifting from conventional prototypes to modern automized systems with travel rate at high speed. Major segments that ease the service of these modern rail transit systems are tracks, signalling and communication, and rolling stock. Among these segments, tracks are said to contribute towards ensuring speed and safety [8]. Tracks are therefore subjected to various types of maintenance to ensure their performance based on critical inspection varying from days to years. Maintenance is classified according to the severity of operations, frequency of inspections, and nature of work involved as preventive maintenance strategy, corrective maintenance strategy and periodic maintenance strategy. The various maintenance strategies involve functionality check of each component associated with the railroad system (Track). Regular preventive/ scheduled maintenance is planned to fit into the defined cycles which includes inspections and minor intervention works like lubrication of fastening systems, cleaning of premises etc. Conditional preventive maintenance aims to restore condition of track back within permissible tolerances involving track geometry component defects. Non-scheduled/ corrective maintenance carries out all unexpected actions involved in an emergency/necessary repair of components along the railroad systems such as rail fracture, defect in welds/joints, broken fastening systems [9]. Furthermore, corrective maintenance is categorized into two types namely temporary corrective maintenance and final corrective maintenance. The above-mentioned maintenance inspections have been followed on a cyclic basis namely weekly, fortnight, half-yearly and Yearly intervention.

Every Indian metro is planned with service runs from 6:00 h to 10:00 h (Morning) on an average that accommodates commuters throughout the transit. The maintenance activities along the railroad system are usually performed in the pace between 12:00 h to 4:00 h (Evening). At times these predefined slots are occupied by the other segments namely Signalling & communication, Rolling stock for their respective maintenance activities which also ends up together in a combined maintenance sequence as given in Fig. 8. Therefore, the periodic maintenance interventions may not be possible within this time period. In contrast certain track segments may not be in need to be serviced/maintained even after rendering service run of fifteen days (Fitness as prescribed by Indian Metro Rail Parameters) without any major interruption which might lead to track renewal. In this case there is an unwanted/ unnecessary utilization of resources to overperform the maintenance activities repeatedly which results in overutilization of resources. In mere future every Indian metro rail system will perform service runs day and night simultaneously connecting cities and states. Therefore, there arises a need to schedule railroad maintenance activities in between the simultaneous service runs and maintenance schedule for further expanded rail network [10].

This study aims to propose a train re-routing schedule in accordance with the maintenance schedule interventions along ten stations based on fuzzy logic reasoning and multilevel machine learning algorithmic approach for Indian metro rail systems to provide uninterrupted service in future.

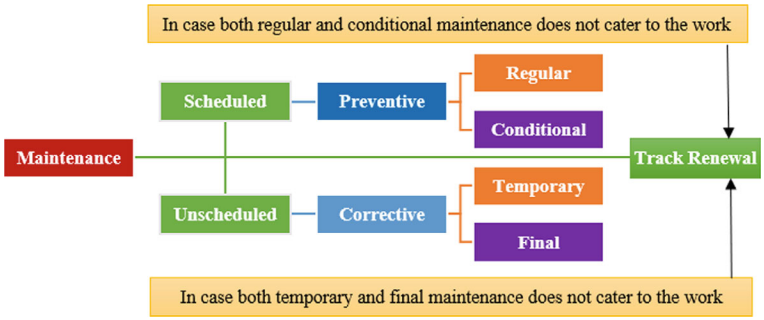


Fig. 8 Types of track maintenance interventions

5.1 Trail of History

In order to better expedite out the process of building a rerouting model along with the track maintenance schedule there has been n number of research works carried out in an international perspective [11]. Few research works that stands close in connection with the proposed research work is tabulated in Table 1 as follows:

Most of the above summarized research works have been concentrated on scheduling of trains based on maintenance interventions and rescheduling of maintenance interventions based on track possession time. Mixed integer programming methods were commonly employed to resolve the boundary constraints. Few other studies utilized heuristics, logistics and mathematical programming models to propose rescheduled tasks be it train timetable or maintenance schedule. The major drawback of the above said methods is that they cannot automatically prioritize the constraints based on severity of probability [22]. Therefore, the proposed research work incorporates neural network model to prioritize the threshold levels of severity and perform re-routing simultaneously to the nearby node lane without much delay in commutation time to avoid cancellation or blockade of path until the maintenance intervention is complete enough to equip regular traffic loads along both service lanes simultaneously.

5.2 Functionalities Prioritization Using Neural Network Model

Most of the recently constructed metro stations in India is characterized by two rail lanes for to and fro movement of trains. Train movements occurs mostly during the daytime for passenger commutation and therefore the track maintenance activities are scheduled during the evening time which goes on till next morning i.e. from 12:00 h to 4:00 h. In mere future with expanding lanes and distance there will be an urge in demand of trains connecting cities to states. There may be conflicts arising

Table 1 Existing Literature database and inferences

Authors	Proposed research	Computing platforms	Key inferences
Mostafa Bababeik et al.	Adjustment of train timetable along single track for maintenance operations while including uncertainties encountered the operational constraints	Mathematical Programming model	Inclusion of buffer time for uncertainties reduces delay in commutation time through the timetable
Zhou Su	Determination of optimal time schedule for maintenance activities and crew	Arc Routing Problem/ Logistics	Flexibility of maintenance time resulted in lower maintenance cost including penalities associated with track possession time
Miwa [12]	Determination of optimal modular maintenance schedule including risk assessment parameters according to ISO 55000	Mathematical Programming model	Determines the number of maintenance work team for an activity while also providing optimal position of material and teams depot along rail line
Albrecht et al. [8]	Development of integrated timetable for a rail network incorporating both train and track maintenance schedule	Meta-Heuristics	Approximately 36% of the generated alternate timetables were found to be much better than the current practice
Besinovic et al. [11]	Development of hierarchical framework for a stable train time table design with delay recovery	Integer Linear Programming (ILP)	A high-quality timetable was produced within 3 to 15 iterations depending on the train line plans
D'Ariano et al. [13]	Optimization of train sequencing based on routing and timing decisions for short term maintenance works	Mixed Integer Linear Programming (MILP)	Flexibility in train routing helps to compute a compact schedule where the pairing of maintenance activity takes place
Luan et al. [14]	Simultaneous optimization of train routes and work time of preventive maintenance tasks	Lagrangian relaxation	Effectiveness of about 25% by 60 s is obtained through integrated scheduling method
Macedo e al. [15]	Scheduling preventive railway maintenance activities by taking available resources into account and anticipating expected failures	Mixed Integer Linear Programming / Variable Neighborhood Search (VNS)	CPLEX solver provides an average computational time of 1047.7 s while the VNS approach required only 1.2 s on an average to arrive at the best solution

(continued)

Table 1 (continued)

Authors	Proposed research	Computing platforms	Key inferences
Santos et al. [16]	Formulation of a decision rules model (DRM) for scheduling heavy maintenance interventions to minimize travel cost and time spend on maintenance	Decisions Rule Model (DRM)	The DRM model was found to be more effective in the areas with regular work intensity where there is reduction in the unit maintenance cost per year
Wang et al. [17]	Scheduling overnight high-speed trains to evict the conflict between the evening track maintenance schedule and to improve the timetable and maintenance plan	Mixed Integer Linear Programming (MILP)	Simultaneous track maintenance plan and path assumption for overnight trains is predicted in advance based on the passenger demand
Liden [18]	Optimal planning and scheduling railway maintenance windows and train traffic	Mixed Integer Linear Programming (MILP)	The integrated approach has credibly led to savings on maintenance cost by 11–17%
Louwerse [19]	Adjustment of schedule of a railway line during a partial or complete blockade due to unexpected disruptions	Integer Programming Formulation	The number of trains that can be operated with delay in time is reflected in the disposition timetable which is found to be in acceptance with the real time disruption management
Arenas et al. [9]	Schedule rearrangement of trains to integrate the maintenance trains against unplanned maintenance activities	Mixed Integer Linear Programming (MILP)	The resulting timetables were found to be better in terms of overall delay reduction than the one obtained by emulating the current practice
Higgins [20]	Determine the allocation of maintenance activities and crews to minimize the disruption of scheduled trains and work completion time	Integer Programming/ Tabu search heuristics	The final schedule achieves 8% reduction in interference delay of trains, 7% reduction in average work completion times
Su et al. [21]	Determine effective maintenance strategy by creating a model for arriving at an optimal schedule for track maintenance irregularities	Mathematical Programming Model	The standard deviation of the surface irregularities is kept at a satisfactory level to achieve effective maintenance

in scheduling midnight trains as well as maintenance activities which may overlap at certain junctures [13]. Moreover, there is a possibility of looping or re-routing the trains at segments where any disruptions have occurred in the context of foreign metros which is not highly possible in Indian metro rail systems. At present there exists turnout sections/emergency loop provision to shift the trains to neighboring tracks in case of emergency interventions. The looping facility is planned only at the turnout junctures which are placed at a distance of 3 to 4 kms apart from every station. In case if a track is shut down for emergency repair before the turnout facility the train operators or the track managers have to shut down the service for minutes or hours and this might reflect in overall delay in schedule [20]. The neighboring track may itself be engaged in its own preplanned service runs and there may not be any possible time space to include the other lane trains to avoid delayed run. In order to overcome the difficulty in rerouting of trains to the simultaneous lane this study adopts rerouting strategy even before it reaches the looping space. At times the preplanned maintenance activities are overperformed on certain track segments which does not require repeated maintenance actions and is considered to be fit for certain longer period. In order to bypass this segment without engaging the segment for unnecessary utilization of resources the maintenance activities are prioritized based on the track segments threshold levels [18]. As discussed, the present work concentrates on the experimental platform which includes two lanes holding twenty stations that accommodates preplanned maintenance strategy as given in Tables 2 and 3.

The existing schedule of trains that runs along the forward lane Track A is represented in Table 1. MTA1 represents the train number which can be expanded as Metro

Table 2 Timetable of existing trains along track A

Train Number at Track A	Train Time at which they reach different stations along Track A										
		AT 1	AT 2	AT 3	AT 4	AT 5	AT 6	AT 7	AT 8	AT 9	AT10
MTA1		7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30
MTA2		7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30	8:35
MTA3	T	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30	8:35	8:40
MTA4		8:00	8:05	8:10	8:15	8:20	8:25	8:30	8:35	8:40	8:45
MTA5		8:05	8:10	8:15	8:20	8:25	8:30	8:35	8:40	8:45	8:50
Delayed Run time		7:45	7:50	7:55	8:00	8:30	8:35	8:40	8:45	8:50	8:55

Table 3 Timetable of existing train along track B

	Train Time at which they reach different stations along Track B										
Train Number at Track B	T	BT 1	BT 2	BT 3	BT 4	BT 5	BT 6	BT 7	BT 8	BT 9	BT10
MTB1		8:30	8:25	8:20	8:15	8:10	8:05	8:00	7:55	7:50	7:45
MTB2		8:35	8:30	8:25	8:20	8:15	8:10	8:05	8:00	7:55	7:50
MTB3		8:40	8:35	8:30	8:25	8:20	8:15	8:10	8:05	8:00	7:55
MTB4		8:45	8:40	8:35	8:30	8:25	8:20	8:15	8:10	8:05	8:00
MTB5		8:50	8:45	8:40	8:35	8:30	8:25	8:20	8:15	8:10	8:05
Delayed Run time		8:55	8:50	8:45	8:40	8:35	8:30	8:25	8:20	7:50	7:45

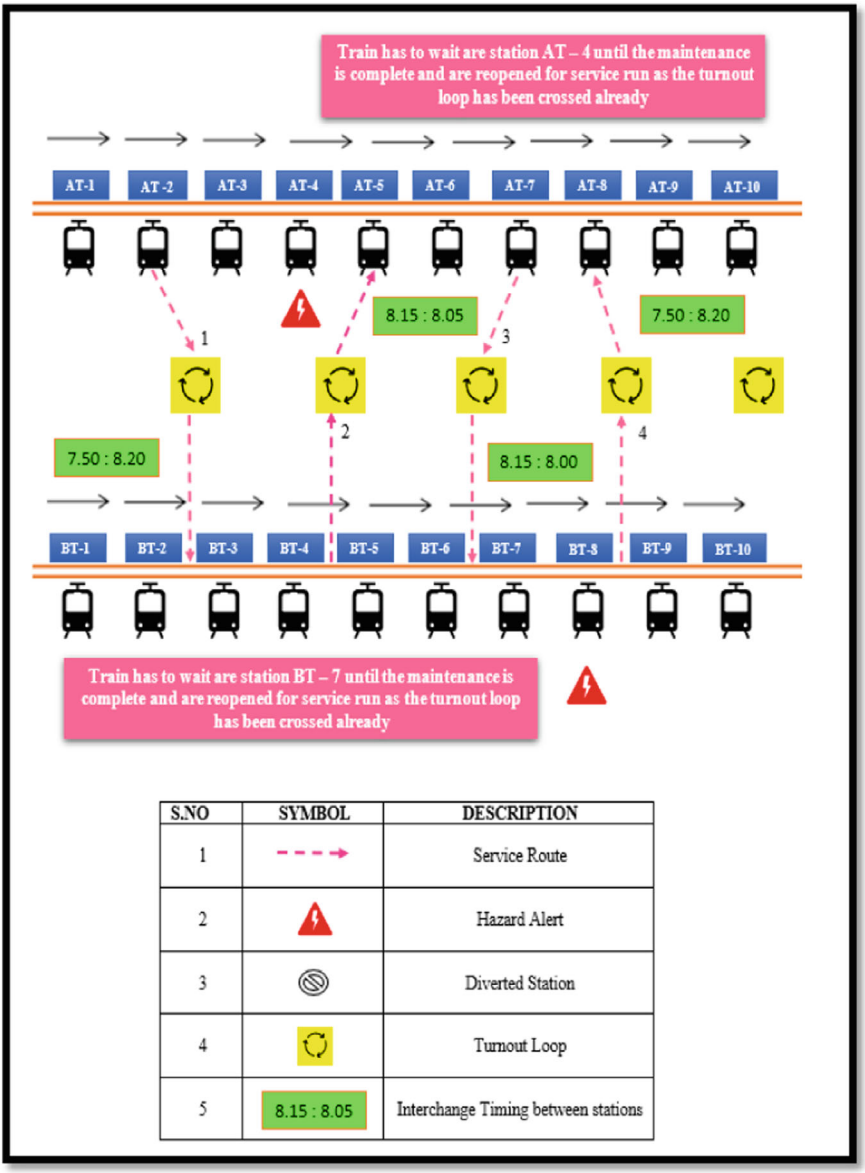
train A1 where AT1, AT2, AT3 represents the station numbers which are consecutively numbered upto AT10 along the forward lane track A where T represents the time at which the respective trains reach the stations. The timetable is pre- scheduled for regular peak hours from 7:45 h to 8:30 h during daytime respectively. The initial start time is represented along green color and the time it reaches the terminal station is represented in orange color. The time space required for maintenance intervention is represented in blue color. From Fig. 1 and Table 1 the maintenance intervention at station AT5 results in a delayed run time and the rescheduled run time is represented in yellow color. The time taken to reach successive stations is said to be 5 min at the minimum and 7 min at the maximum. The time taken to restore the track for service run is assumed to be about 25- 30 min. The overall delay in runtime of train MTA1 is said to be 25 min i.e. from 8:30 h to 8:55 h in the rescheduled time at the minimum or 8:57 in the rescheduled time at the maximum. The existing schedule of trains that runs along the backward lane Track B is represented in Table 2. MTB1 represents the train number as Metro train B1 where BT1, BT2, BT3 represents the station numbers which are consecutively numbered upto BT10 along the backward lane track B. The timetable is scheduled for regular peak hours from 7:45 h to 8:30 h during daytime respectively. The initial start time is represented along green color and the time it reaches the terminal station is represented in orange color [14]. The time space required for maintenance intervention is represented in blue color.

From Figure 9 and Table 1 the maintenance intervention at station AT5 results in a delayed run time and the rescheduled run time is represented in yellow color. The time taken to reach successive stations is said to be 5 min at the minimum and 7 min

at the maximum. Turnouts are positioned in the mid of every two successive stations i.e after every even station number. The time taken to restore the track for service run is assumed to be about 25–30 min. The overall delay in runtime of train MTA1 is said to be 25 min i.e from 8:30 h to 8:55 h in the rescheduled time at the minimum or 8:57 in the rescheduled time at the maximum.

Figure 9 provides a pictorial representation of the forward and backward lane maintenance interventions and the intercross turnout section timings that is to be followed at most of the metro rail systems in practice across the nation. Usually, trainsets along the track A are withdrawn from service run until the maintenance intervention is complete during daytime. In case of any emergency maintenance breakdown of the train or tracks, the trains are rerouted at the looping area along the turnout sections. Therefore, Fig. 1 provides a brief insight about how the tracks can be rerouted manually for travel along track A and B simultaneously assuming the maintenance interventions also happen during daytime as well as nighttime. The entire stretch of track A and B engulfs 5 turnouts throughout the study. The looping along the turnouts is numbered as 1,2,3 and 4 respectively. Along loop1 where the emergency maintenance intervention should take place at AT4 the trains have to reroute long the turnout section at station AT2 to crossover to BT3. The time of intercrossing at AT2 is 7:50 h and the time at which it reaches BT3 will be 8:05 h i.e. the pre-scheduled train along the track B would reach BT3 by 8:20 h and therefore there is a time space of about 15 min. At loop 2 the train shifts to track A after BT4 through the turnout section to reach AT5. The time of intercrossing at BT4 is 8:15 h and the time at which it reaches AT5 would be 8:30 i.e. the pre-scheduled train along the track A should have crossed by 8:05 h which was not possible due to the maintenance intervention. At loop 4 the train shifts to track A through the turnout section from BT8 to AT8. The time of intercrossing at BT8 is 7:50 h and the time to reach AT8 is 8:05 h i.e. the prescheduled train would cross AT8 at 8:20 h and the time space available for next train is maintained as 15 min. At loop 3 the train shifts back from AT7 to BT6 through the turnout section. The time of intercrossing at AT8 is 8:15 h whereas the train reaching BT6 has already passed at 8:00 h from BT7 and reached BT6 at 8:05 h. Therefore, on an average the time available will be 15 min at every intercrossing section to reach the neighboring track stations.

From Tables 3 and 4 the intercrossing schedule along track A and track B is presented. NC represents no change in schedule, ML represents maintenance lane, FL represents free lane. The rescheduled route is highlighted in pink color, unaltered or regular time is represented in grey color, time space available between the rescheduled runs is highlighted in powdered blue color. MTA1 crosses station 2 at 7:50 h and reach station 3 at track B at 8:05 h and it is represented as 8:05A. Further the train crosses station 4 along track B at 8:10 h which is represented as 8:10A. The train gets back to station 5 at 8:20 h and reaches station 10 at 8:45A if there are no issues along track A. From Table 1 it can be visualized that the regular run time along track A at station 8 is 8:20 h which has a tendancy to collapse or overlap with the rescheduled train from track B at station 8 if the assumed service time either exceeds the minimum (5 min) or maximum (7 min) over circumstances (Table 5).



S.NO	SYMBOL	DESCRIPTION
1		Service Route
2		Hazard Alert
3		Diverted Station
4		Turnout Loop
5	8.15 : 8.05	Interchange Timing between stations

Fig. 9 Maintenance routing methodology by neural network

Table 4 Intercrossing schedule for Mta1 at track A

Train Number	Track A									
	1	2	3	4	5	6	7	8	9	10
MTA1	NC	7:50	FL	ML	8:20A	8:25A	8:30A	8:35A	8:40A	8:45A
					8:05	8:10	8:15	8:20A	8:25	8:30
							8:25B	8:20B		
	-	-	-	-	-	-	10-15	⚠	-	-

Table 5 Intercrossing schedule for Mtb1 at track B

Train Number	Track B									
	1	2	3	4	5	6	7	8	9	10
MTB1	8:30	8:25	8:20	8:10A	8:40B	8:35B	FL	ML	NC	NC
			8:05A	8:15	8:10	8:05				
	9:00B	8:55B	8:50B	8:45B						
	-	-		35	-	-	-	-	-	-

6 Conclusions

With the creating acknowledgment of IoT gadgets, severa particular offices are growing new situation to control and control the cooperation among IoT and clients. IoT applications are explicitly used to catch and technique the sensor realities and to computerize and execute assignments. In acclaimed, the ones IoT frameworks give the way to IoT gadgets to converse with every single one of a sort and to give constant assessment to clients. Because of the sizeable design of bundles in IoT, the ones frameworks assortment in their point target market and uses; be that as it may, they all need to in spite of the fact that cling to explicit programming and compositional standards to be feasible. As a benchmark, fine-grained get legitimate of admittance to control and verification for all elements (client, gadgets) with individual authorizations and personalities at each layer of an IoT arrangement should be completed in the present IoT stage. We be cognizant that also research in IoT insurance and protection remains shockingly wanted in subjects which incorporate alter safe equipment, IoT programming program assessment, IoT gadget/diversion revelation, halting records spillage, and well-known weakness disclosure. Hence through continuous data monitoring of the oil pipeline and metro track functioning performance through IoT, a well-decision-making model have been developed. Neural network model investigated on the oil pipeline have proved enough pre-emptive decision making on the risk estimation. Similarly, the metro track timing schedule self-automatic routing and scheduling have been designed using neural network and proved efficient model through analysis.

References

1. Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M.: Deep learning for iot big data and streaming analytics: a survey. IEEE Commun. Surv. Tutorials **20**(4), 2923–2960 (2018). https://doi.org/10.1109/COMST.2018.2844341
2. Wang, J., Hu, J., Min, G., Zomaya, A.Y., Georgalas, N.: Fast adaptive task offloading in edge computing based on meta reinforcement learning. IEEE Trans. Parallel Distrib. Syst. **32**(1), 242–253 (2021). https://doi.org/10.1109/TPDS.2020.3014896
3. Luong, N.C., Hoang, D.T., Gong, S., Niyato, D., Wang, P., Liang, Y., Kim, D.I.: Applications of deep reinforcement learning in communications and networking: a survey. IEEE Commun. Surv. Tutorials **21**(4), 3133–3174 (2019). https://doi.org/10.1109/COMST.2019.2916583
4. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., Jue, J.P.: All one needs to know about fog computing and related edge computing paradigms: a complete survey. J. Syst. Architect. **98**, 289–330 (2019). https://doi.org/10.1016/j.sysarc.2019.02.009
5. Bhaskaran, P.E., Chennippan, M., Subramaniam, T.: Future prediction & estimation of faults occurrences in oil pipelines by using data clustering with time series forecasting. J. Loss Prev. Process Ind. **66**, 104203 (2020). https://doi.org/10.1016/j.jlp.2020.104203
6. Priyanka, E.B., Thangavel, S.: Decision making based on machine learning algorithm for identifying failure rates in the oil transportation pipeline. 2020 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, pp. 914–919 (2020)
7. Bhaskaran, P.E., Maheswari, C., Thangavel, S., Ponnibala, M., Kalavathidevi, T., Sivakumar, N.S.: IoT Based monitoring and control of fluid transportation using machine learning. Comput. Electr. Eng. **89**, 106899 (2021). https://doi.org/10.1016/j.compeleceng.2020.106899
8. Albrecht, A.R., Panton, D.M., Lee, D.H.: Rescheduling rail networks with maintenance disruptions using problem space search. Comput. Oper. Res. **40**(3), 703–712 (2013). https://doi.org/10.1016/j.cor.2010.09.001
9. Arenas, D., Pellegrini, P., Hanafi, S., Rodriguez, J.: Timetable rearrangement to cope with railway maintenance activities. Comput. Oper. Res. **95**, 123–138 (2018). https://doi.org/10.1016/j.cor.2018.02.018
10. Bababeik, M., Zerguini, S., Farjad-Amin, M., Khademi, N., Bagheri, M.: Developing a train timetable according to track maintenance plans: a stochastic optimization of buffer time schedules. Transp. Res. Procedia **37**, 27–34 (2019). https://doi.org/10.1016/j.trpro.2018.12.162
11. Besinovic, N., Goverde, R.M., Quaglietta, E., Roberti, R.: An integrated micro–macro approach to robust railway timetabling. Transp. Res. Part B: Methodol. **87**(1), 14–32 (2016). https://doi.org/10.1016/j.trb.2016.02.004
12. Miwa, M.: Mathematical programming model analysis for the optimal track maintenance schedule. Quart. Report of RTRI **43**(3), 131–136 (2002). https://doi.org/10.2219/rtriqr.43.131
13. D'Ariano, A., Meng, L., Centulio, G., Corman, F.: Integrated stochastic optimization approaches for tactical scheduling of trains and railway infrastructure maintenance. Comput. Ind. Eng. **127**, 1315–1335 (2019). https://doi.org/10.1016/j.cie.2017.12.010
14. Luan, X., Miao, J., Meng, L., Corman, F., Lodewijks, G.: Integrated optimization on train scheduling and preventive maintenance time slots planning. Transp. Res. Part C: Emerg. Technol. **80**, 329–359 (2017). https://doi.org/10.1016/j.trc.2017.04.010
15. Macedo, R., Benmansour, R., Artiba, A., Mladenovic, N., Urosevic, D.: Scheduling preventive railway maintenance activities with resource constraints. Electron. Notes Discrete Math. **58**, 215–222 (2017). https://doi.org/10.1016/j.endm.2017.03.028
16. Santos, R., Teixeira, P.F., Antunes, A.P.: Planning and scheduling efficient heavy rail track maintenance through a Decision Rules Model. Res. Transp. Econ. **54**, 20–32 (2015). https://doi.org/10.1016/j.retrec.2015.10.022
17. Wang, D., Zhan, S., Peng, Q., Zhou, W.: integrated overnight train scheduling and maintenance planning for high-speed railway lines. Transp. Res. Record **2675**(3), 222–237 2021 https://doi.org/10.1177/2F0361198120968830

18. Liden, T.: Coordinating maintenance windows and train traffic: a case study. Public Transport **12**, 261–298 (2020). https://doi.org/10.1007/s12469-020-00232-2
19. Louwerse, I., Huisman, D.: Adjusting a railway timetable in case of partial or complete blockades. Eur. J. Oper. Res. **235**(3), 583–593 (2014). https://doi.org/10.1016/j.ejor.2013.12.020
20. Higgins, A.: Scheduling of railway track maintenance activities and crews. J. Oper. Res. Soc. **49**(10), 1026–1033 (1998). https://doi.org/10.1057/palgrave.jors.2600612
21. Su, Z., De Schutter, B.: Optimal scheduling of track maintenance activities for railway networks. IFAC-Papers Online **51**(9), 386–391 (2018). https://doi.org/10.1016/j.ifacol.2018.07.063
22. Consilvio, A., Di Febbraro, A., Sacco, N.: A modular model to schedule predictive railway maintenance operations. In 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE, 426–433 (2015). https://doi.org/10.1109/MTITS.2015.7223290

Forecasting of Lake Level by Soft Computing Approaches

Vahdettin Demir, Mehmet Ali Tamer, and Serdar Carbas

Abstract To ensure the sustainability and management of water resources, regularly monitoring the water levels in lakes, rivers, basins, dam reservoirs, etc. is a very important engineering task. Our freshwater resources are gradually decreasing due to the destruction of freshwater resources and climate change. For this reason, monitoring, modelling, and researching of freshwater resources, especially lakes, are increasingly important issue for nowadays. In this chapter, soft computing approaches are used to forecast of lake water levels at Beyşehir Lake, located in the central part of Turkey. To do this, three artificial neural network algorithms (Multilayer, Radial Basis and Generalized Regression), two heuristic algorithms (Model 5-Tree and Multivariate Adaptive Regression Spline), and a Support Vector Machines containing three different functions (Radial, Polynomial, and Linear) are used. In addition to being models used successfully in hydrological modelling of civil engineering, the changes in modelling performance with the number of iterations, kernel functions, optimization algorithms, and data input sets that constitute the internal dynamite of the techniques are investigated. The attained results show that through these multiple parameters, radial basis artificial neural networks are the most successful when compared with mean absolute error, root mean square error, coefficient of determination, Taylor diagrams and Violin plots.

Please note that the AISC Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

V. Demir (✉)
Department of Civil Engineering, KTO Karatay University, Konya, Türkiye
e-mail: vahdettin.demir@karatay.edu.tr

M. A. Tamer
Graduate Education Institute, KTO Karatay University, Konya, Türkiye
e-mail: mehmet.ali.tamer@ogrenci.karatay.edu.tr

S. Carbas
Department of Civil Engineering, Karamanoglu Mehmetbey University, Karaman, Türkiye
e-mail: scarbas@kmu.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_6

Keywords Artificial neural network · Support vector machines · Heuristic algorithms · Lake-level forecasting

1 Introduction

Civil engineering is concerned with many applications such as superstructure and infrastructure development, transportation and traffic, balancing of dynamic and mechanical systems, water resource planning and management, ground reinforcement and improvement [1]. In general, all applications and designs related to the living environment are within the scope of civil engineering. Hydraulic engineering, a sub-discipline of civil engineering, is the field where the effect of machine learning techniques and hyperparameters on problem solving is most researched, such as planning the management of circulating water [2]. Water is an important and indispensable resource for all life forms on Earth to survive. Also, water is a very necessary resource not only for human use, but also for the integrity of the ecosystem, the development of the economy, ensuring national security and energy production [3]. The water has played a very important role in the emergence and development of civilizations since the existence of human beings. Since the beginning of civilizations, all civilizations that chose their first settlements near the water resources have become advantageous in development as they learned to use water in industrial and agricultural areas [4]. Throughout the time, humanity, in its diverse pursuits, learned to harness water, recognizing its significance for sustaining life on Earth. However, as societies increasingly exploit water resources, individuals have become cognizant of the shifts and depletions occurring in global water reservoirs. Prompted by these observations, investigations into the causes of water scarcity have ensued, revealing climate change as a primary contributor to the fluctuations in water availability [5–8].

The most important phenomena that may occur as a result of climate change are basic climatic and hydrological variables such as precipitation, temperature, flow, evapotranspiration and lake water level [9, 10]. Climate changes, the greenhouse effect in the atmosphere, climatic and hydrological aspects, and tectonic changes are regarded to be the primary causes of lake water level fluctuations. Too low or too high lake water levels cause great economic losses and irreparable natural events in the lake and its surroundings [11, 12]. These fluctuations in the masses of water, which is our source of life, have increased the research on the quantity and quality of existing water resources, especially in the field of earth sciences and civil engineering, and water level trends and water level modelling of lakes have gained great importance [13–15].

A detailed literature research on the water level of lakes was conducted in the Web of Science database with the keywords "*lake level, machine learning*". As a result of the examinations conducted in March 2024, it was observed that there were 590 academic studies in total, including the relevant keywords, and 464 of these studies consisted of articles and 2 of them consisted of book chapters. These studies were

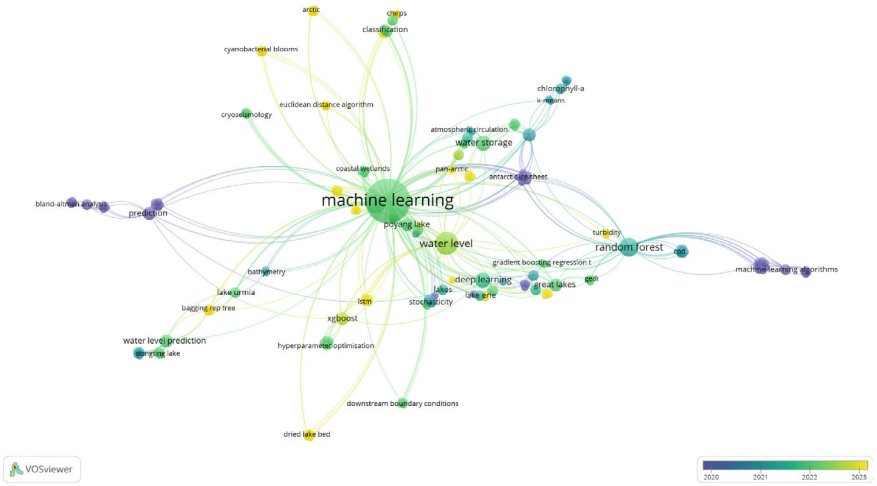


Fig. 1 Connection graph of keywords

limited to 3 common keywords and linked through VOSviewer V1.6.20 software [16, 17], and the connection to the keywords are given in Fig. 1.

In Fig. 1, it is observed that the relationship map of keywords, machine learning and hyper parameter studies have a central position in lake water level prediction. In addition, machine learning techniques such as deep learning, Long Short-Term Memory (LSTM), XGBoost, gradient boosting regression and random forest are among the models preferred by researchers in recent years. Upon closer examination of studies on this subject, it is evident that, Altunkaynak [18] the change of Van Lake water level over time was examined with the help of artificial neural networks method. He stated that artificial neural networks are simpler and more reliable than other traditional methods (AR, MA, ARMAX). Yarar and Onüçyıldız [19] tried to predict the water level changes of Lake Beyşehir using an artificial neural network, using data between 1962 and 1990 measured by DSI, to investigate the potential of using Artificial Neural Networks in modeling the lake water level. As a result of the study, it was stated that the water level of Lake Beyşehir can be easily modeled by adapting the water balance equation to the Artificial Neural Networks method. Additionally, the level fluctuations of the Lake Beyşehir were calculated using adaptive neuro-fuzzy inference system (ANFIS), artificial neural networks (ANN), and seasonal autoregressive integrated moving average (SARIMA) [20]. Karimi et al. [21] used data-oriented GEP and ANFIS techniques, which are two different machine learning approaches, in their study using data covering the period between 1972 and 2003 to estimate water level changes in Lake Urmia, located in the northwest of Iran. They stated that GEP was slightly superior to other applied models in each of the prediction intervals. Researchers also emphasized that GEP and ANFIS techniques can be used to estimate lake water levels. Shiri et al. [22] analyzed the observation data for the period between 1972 and 2003 with two different prediction models in order to

predict the daily water level changes of Lake Urmia, located in the northwest of Iran. Researchers compared the extreme learning machines (ELM) method with genetic programming (GP) and ANN methods to evaluate the accuracy of the prediction data obtained from the analysis. It is stated that the ELM technique was more successful than GP and ANN models in predicting the water level of Lake Urmia. Zhu et al. [23] aimed to estimate the monthly water level of 69 lakes in Poland using lake water level data covering the period 2006–2019. In the prediction phase, they used two different machine learning models, Feed Forward Neural Network (FFNN) and Deep Learning (DL) techniques. As a result of the study, they stated that generally good results were obtained from both models and that the Deep Learning model was not superior to the feedforward neural network model. Wang and Wang [24] used six different machine learning algorithms to predict the water level in Lake Erie. As a result, it was observed that multiple linear regression and M5P model tree algorithms had the highest degree of accuracy in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Peprah and Larbi [25] conducted a study to evaluate the efficiency and applicability of different ANN techniques in the change of lake water level in the Volta Lake basin. Satellite altimeter data provided by the United States Department of Agriculture, covering the years 1992–2019, was used in the study. Researchers who statistically evaluated each model technique and its results stated that all models successfully predicted the lake water level (LWL). A monthly average lake water level estimate for Lake Michigan was attempted by Çubukçu et al. [26]. Three distinct artificial neural networks (ANNs) were used in the study to estimate monthly average lake water level data of Lake Michigan for the years 1981 to 2020 (Multilayer ANN, Radian Based ANN, and Generalized ANN). The study's findings indicate that Multi-layer ANN is the best effective training method for prediction based on test results. Demir and Yaseen [27] used lake water level data for the period between 1918 and 2020 to predict fluctuations that may occur in the lake water level for five large lakes in North America. They developed neuro-computer intelligence models (M5-Tree, MARS, LSSVR). As a result of their analysis and studies, the researchers stated that the P-LSSVR model is a powerful model in lake water level prediction and is superior to and an alternative to other neuro-computational intelligence models used in the study. Özdemir et al. [28] emphasized that changes in meteorological and hydrological parameters due to global climate change and intensive water use will cause large fluctuations in lake water levels. In this study, researchers examined six different types of algorithms to estimate the water level in lakes. As a result of the predictions made, they stated that the deep learning model has the highest accuracy. Zakaria et al. [29] used three different machine learning algorithms (XGBoost, Multi-layer Perceptron (MLP), and LSTM) to estimate the water level of the Muda River in Malaysia. They were modeled and tested with the help of limited daily water level and meteorological data from 2016 to 2018 and evaluated with different performance scenarios. As a result of the study, it was observed that the MLP model performed better than the LSTM and XGBoost models.

In this chapter, lake water level data (m) from Lake Beyşehir, Turkey's largest freshwater lake, were collected between 1950 and 2022, and the results of six different

techniques were compared. Used methods include three ANN algorithms (Multilayer, Radial Basis, and Generalized Regression), two heuristic algorithms (Model 5-Tree and Multivariate Adaptive Regression Spline), and a Support Vector Machine with three functions (Radial, Polynomial, and Linear). In modeling, the data set consisting of monthly data was divided into four equal parts, 75% of which was used in the training phase and 25% of which was used in the testing phases to each contain different combinations, and the robustness of the models was compared according to various performance criteria.

2 Material and Method

2.1 Study Area and Data

Lake Beyşehir is located in the south-west of Turkey, between the Konya and Isparta provinces in the region known as the Lakes Region. Lake Beysehir, Turkey's third largest lake, is also the country's biggest freshwater lake. The lake was formed as a result of tectonic and carstic events. Surrounded by mountains with a height exceeding 2000 m, Beysehir Lake is located in a tectonic depression [30]. Figures 2 and 3 display a map indicating the lake's location. As of 11.01.1993, the national park protection area of Lake Beyşehir and its surroundings, which was taken under the status of a National Park, is 82157 hectares. The surface of the lake, which is 1121 m above the sea, has a surface area of 651 km^2 and a length of 45 km. While the widest part of the lake is 26 km, the narrowest part is 14 km. In addition, the deepest part is approximately 10 m. The lake has 22 islands and two beaches [31]. Lake Beyşehir National Park, which is among the most important National Parks of Turkey, hosts many activities such as sports and tourism, while the lake and its surroundings are very rich in terms of living diversity. Hence, it has a high economic value, especially in terms of aquatic products [31]. Lake Beyşehir and its surroundings have also been declared a Drinking and Irrigation Water Reserve protected area and a group A wetland of international importance. Furthermore, this region has been registered as a cultural and historical site [32]. Lake Beyşehir, which is the largest lake in the Lakes Region due to its location, is affected by both Central Anatolia and Mediterranean climates. The lake is fed mainly from the Lake Beyşehir basin (4121 km^2). While the recharge of the Lake Beyşehir Basin is formed by precipitation falling on the lake surface, groundwater recharge and surface flows, its discharge is formed by evaporation, irrigation, discharge, and karstic leaks. The streams feeding the lake are Yenişerbademli, Soğuksu, Sarıöz, Ebülvefa, Çavuş, Eflatunpınar, Kurucaova, Ozan and Üstünler streams water leakage occurs in the sinkholes of the lake. Since the sinkhole on the west shore of the lake is higher than the other sinkholes, water leakage starts when it reaches 1123 m level. While the average leakage is approximately 5 m^3/s, it is known that the largest leakage was 29.7 m^3/s in March 1976 [32].

Fig. 2 Location of Konya closed Basin

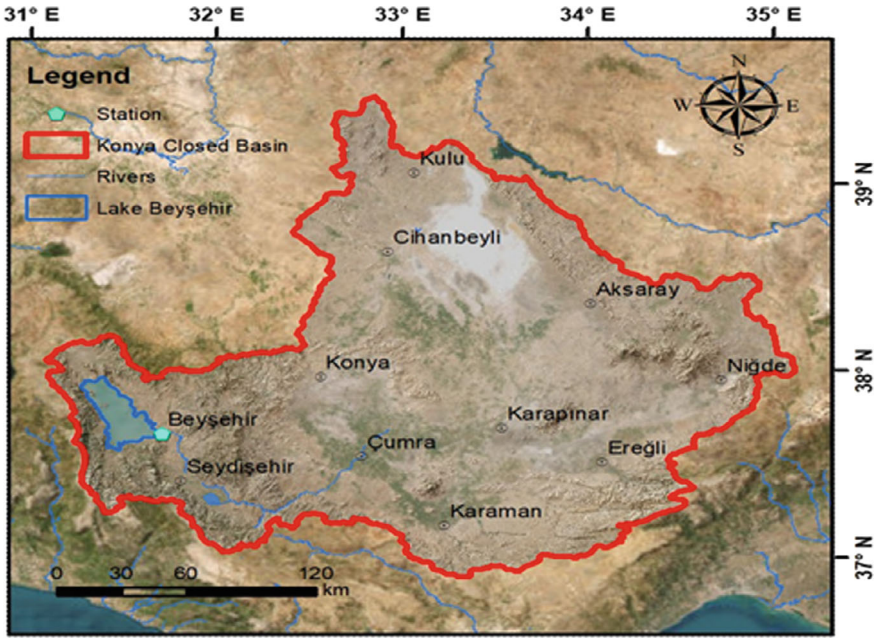


Fig. 3 Study area (Lake Beyşehir) and lake water observation station

The lake's sinkholes are known to spill water into the Manavgat River. Discharge also occurs from the sinkholes located at Homat and Küre cape, east of Mada Island, around Hacı Arif and Kül Island. The waters of the Gembos Polyesi in the southern region of the lake flow into the Manavgat Stream. This underground flow within the karst formation was intervened by the construction of Derebucak Dam and Gembos Diversion Canal. It has been observed that while the waters directed towards Lake Beyşehir with the construction of channels increase the water level in the lake, there is a decrease in the waters of Manavgat Stream [33].

2.2 Data

The data covering the period between 1950–2022 of Lake Beyşehir. In this study, 876 monthly average lake water level (m) records covering 73 years were used. In the modelling phase of design problem, lake water level data of the past months were used as input data. To determine the number of inputs, correlation, autocorrelation, and partial autocorrelation analyses mentioned in the literature were performed and the relationship between the data at time T and the data between months (T-M) were examined with the help of correlation (Fig. 4), autocorrelation (Fig. 5) and partial autocorrelation graphs (Fig. 6).

In Fig. 4, there are strong relationships between the data of previous months (lag-time) and the data in the time series. The graph shows that this relationship decreases until the 8th month, increases again after the 8th month (Fig. 5), but decreases again after the 14th month and these trends continue periodically. For this reason, the data up to the 14th month are accepted as input data in our study. Additionally, when the partial autocorrelation graph is examined (Fig. 6), 14 inputs data from the 1st month to the 14th month were used in the study, since the partial autocorrelation values were entered within the confidence limit for the second time at the 14th input. Output data (lake water level measurement data) representing moment T were estimated by going back one month from moment T.

The monthly data is divided into four equal parts, M1, M2, M3 and M4, considering the chronological order. The 75% of this separated data is used in the training phase and the remaining 25% is used in the testing phase to create data packages called P1, P2, P3 and P4, each with different combinations. The information about our monthly data sets that make up the P1, P2, P3 and P4 data packages is given in Table 1.

In Table 1, for example, the P1 data package, M1 data constitutes the testing phase, while M2, M3 and M4 data constitute the training phase. The time series graph of the lake level is given in Fig. 7.

Statistical information for all training and test data used in our modelling is given in Table 2.

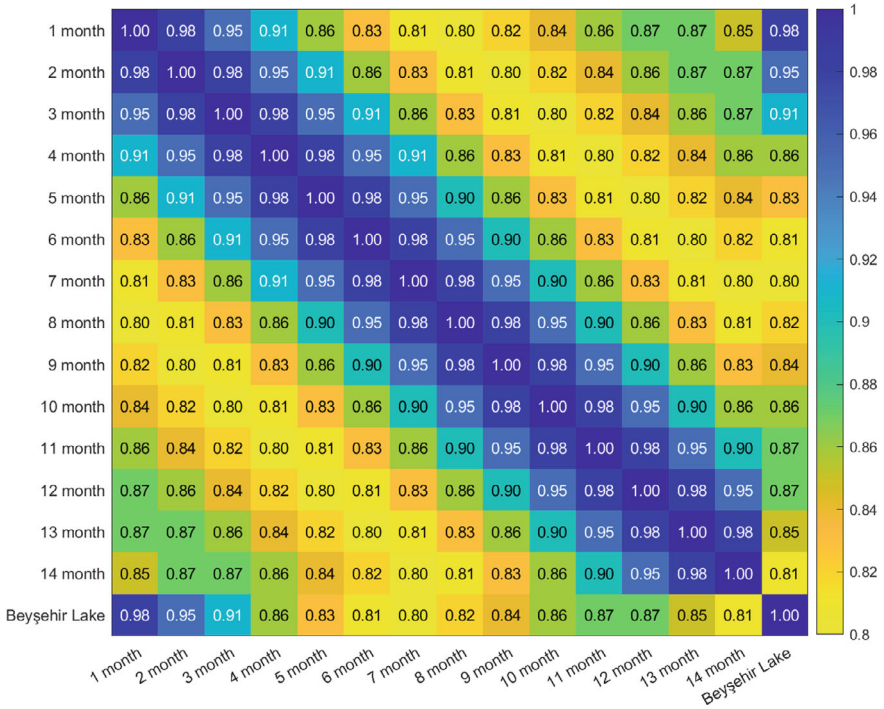


Fig. 4 Correlation graph of Lake Beyşehir

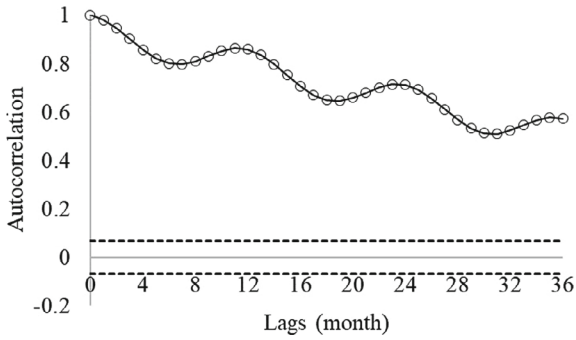


Fig. 5 Autocorrelation graph of data from Lake Beyşehir

2.3 Artificial Neural Networks (ANN)

ANNs are computer systems developed by taking the human brain as an example and designed by adopting the working logic of the biological nervous system. These nervous systems contain neurons, just like in humans, and these neurons are connected to each other in various ways through weighted connections, forming parallel and distributed networks with unique memories. These networks can learn,

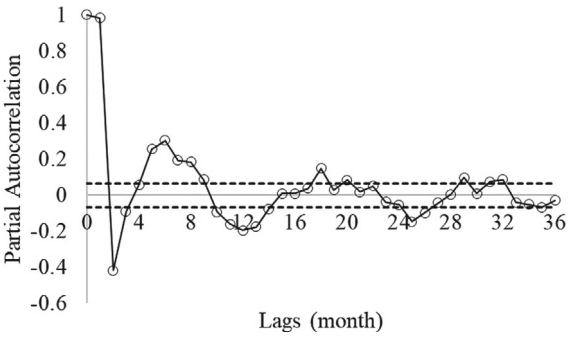


Fig. 6 Partial autocorrelation graph of data from Lake Beyşehir

Table 1 Information about data packets

P_1 %25-Test Data %75-Training Data	M1 (Test Data) (1950/January) (1967/November)	M2 (Training Data) (1967/December) (1985/October)	M3 (Training Data) (1985/November) (2003/September)	M4 (Training Data) (2003/October) (2021/August)
P2 %25-Test Data %75-Training Data	M1 (Training Data) (1950/January) (1967/November)	M2 (Test Data) (1967/December) (1985/October)	M3 (Training Data) (1985/November) (2003/September)	M4 (Training Data) (2003/October) (2021/August)
P3 %25-Test Data %75-Training Data	M1 (Training Data) (1950/January) (1967/November)	M2 (Training Data) (1967/December) (1985/October)	M3 (Test Data) (1985/November) (2003/September)	M4 (Training Data) (2003/October) (2021/August)
P4 %25-Test Data %75-Training Data	M1 (Training Data) (1950/January) (1967/November)	M2 (Training Data) (1967/December) (1985/October)	M3 (Training Data) (1985/November) (2003/September)	M4 (Test Data) (2003/October) (2021/August)

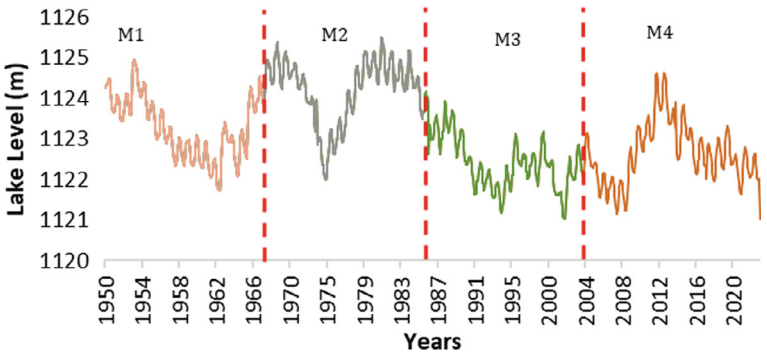


Fig. 7 Lake Beyşehir time series graph and data separations

Table 2 Statistical information with regard to all training and test data

	Data Set	Period	Mean	Minimum	Maximum	Standard deviation	Kurtosis coefficient	Skewness coefficient
	All	(1950/January)–(2021/August)	1123.13	1121.03	1125.50	1.009	−0.862	0.265
P_1	Testing (M_1)	(1950/January)–(1967/November)	1123.26	1121.73	1124.93	0.749	−0.866	0.126
	Training (M2-M3-M4)	(1967/December)–(2021/August)	1123.11	1121.03	1125.50	1.080	−0.974	0.302
P_2	Testing (M2)	(1967/December)–(1985/October)	1124.21	1121.97	1125.50	0.785	0.239	−0.948
	Training (M1-M3-M4)	(1950/January)–(1967/November) (1985/November)–(2021/August)	1122.79	1121.03	1124.93	0.805	−0.434	0.299
P_3	Testing (M3)	(1985/November) -(2003/September)	1122.44	1121.03	1124.17	0.674	−0.300	0.348
	Training (M1-M3-M4)	(1950/January)–(1985/October) (2003/October)–(2021/August)	1123.39	1121.15	1125.50	0.991	−0.963	0.025
P_4	Testing (M4)	(2003/October)–(2021/August)	1122.68	1121.15	1124.63	0.757	−0.116	0.334
	Training (M1-M3-M4)	(1950/January)–(2003/September)	1123.30	1121.03	1125.50	1.035	−1.030	0.071

memorize, and find relationships between data with the help of examples. Therefore, ANNs perform self-learning without the need for traditional programming and algorithm skills. Calculation methods are different from what is known. ANN systems are used in many areas of our lives by adapting to the environment, being stable in the face of uncertainties, tolerant of errors and successful applications despite incomplete information [34–36].

2.3.1 Normalization

In artificial neural networks, certain pre-processing can be applied to the input and output layers of network cells in order to make the training of the data presented to the network more efficient. This pre-processing is called "*Normalization*". The normalization process transforms the raw data into the most suitable data set for our modeling to improve network usage and network input processes. Thanks to normalization, the training of artificial neural networks is carried out faster. There are many normalization rules used in the literature. These are Median, Z-Score, Sigmoid, Min–Max [37].

In this study, Min–Max Normalization (D_Min_Max Normalization) is used. Min–Max Normalization refers to the lowest and highest values that a data can take by normalizing the data linearly. For ANN model, all data were first normalized with values of 0.2 to 0.8, taking into account the operations in the equation specified in Eq. 1 [38].

$$x' = 0.6 \times \frac{x_1 - x_{\min}}{x_{\max} - x_{\min}} + 0.2 \tag{1}$$

Here,

$x\prime$ = Normalized data.

x_1 = Input value.

$x_{\min}$ = The minimum value,

$x_{\max}$ = The maximum value.

2.4 Multilayer Artificial Neural Networks (MLANN)

ANNs that have at least one layer between the input and output layers are called multilayer perceptron (MLP). These networks, which have three different layers (input layer, intermediate layer, output layer), are widely used today because they provide solutions to nonlinear problems. The input layer is the input part of the problem to be solved to the ANN, and the data is not subjected to any processing in this layer. There may be more than one input to the MLP, but each processor has a

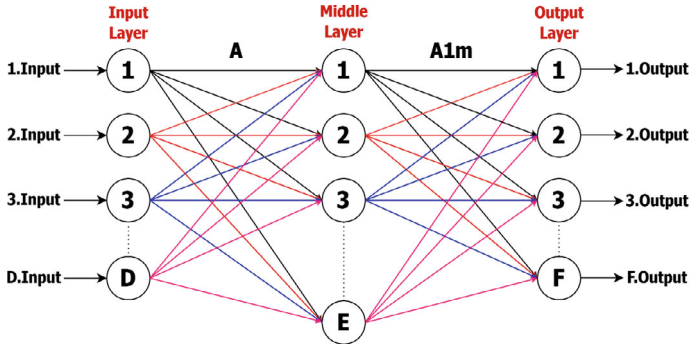


Fig. 8 A three-layer (Multi-layer) ANN structure

single input and a single input. Information passing through the input layer reaches the intermediate layer (hidden layer) and the information is processed in this layer. Generally, if there is no linearity between the inputs and outputs of the problem for which the network is to be learned and the complexity is high, more than one intermediate layer can be used instead of a single intermediate layer. The output layer is the layer where the data processed within the network is transferred to the external environment. Data flow to the multilayer sensor network system is always forward. Figure 8 shows the structure of the MLANN.

In this study, the Levenberg–Marquardt algorithm, which is an advanced algorithm rather than classical algorithms, was used. It plays an important role in solving numerical problems by minimizing nonlinear functions. This algorithm is fast and provides stable convergence. Levenberg–Marquardt algorithm is among the frequently preferred algorithms because it obtains safer and faster results when processing complex curves [25, 39]. In addition, the Logarithmic sigmoid (logsig) function was used in the study, and separate analyzes were made on 100 and 1000 values for the epoch value (iteration). Thanks to iteration, the weights between neurons and layers are adjusted. Whether the training phase ends in a model depends on reaching the maximum number of iterations or the minimum error value to be reduced [40].

2.5 Radial Based Neural Network (RBANN)

The Radial Based ANNs model was created by modeling the action-response behavior of neurons in the nervous system of humans. Radial basis functions are used in areas that will solve problems such as estimation, function approximation and curve fitting, as a new alternative to MLP, due to their features such as shorter training time (faster) than multilayer perceptron and reaching the best solution without getting stuck in local minima [36, 41, 42].

RBANN is a feed-forward network with a 3-layer structure (input, hidden, output). The most basic feature that distinguishes this network from other ANNs is the use of radial based activation functions in the network. While the output of this network is linear, its inputs are nonlinear. The processes that raw data goes through in RBANN are as follows, the data coming to the input layer consisting of source nodes is transmitted to the hidden layer without being multiplied by the weight values, unlike the processes in multi-layer sensors. The input layer is where the network's relationship (connection) with the environment begins. The data passing from the input layer to the hidden layer (second layer) is subject to radial basis transfer functions and a nonlinear constant transformation. While a linear and adaptive transformation occurs from the hidden layer (second layer) to the output layer, the transfer signal applied to the input layer is responded to by the output layer. In this study, purelin function was used as the activation function in the RBANN model. Purelin function is an activation function in which neuron outputs change linearly according to the change in neuron inputs. The dynamic variation range of this function is in the [−1, 1] range [43, 44].

2.6 *Generalized Regression Neural Network (GRANN)*

The Generalized Regression ANN proposed by Donald Francis Specht (1991) is a special case of radial basis networks [45]. As the amount of data in the training set increases, the error value in the predictions in GRANN approaches zero, while the number of transactions performed in the network increases. GRANN is widely used in estimating problems with continuous variables, similar to standard regression techniques [46–48]. Figure 9 shows the structure of the Generalized regression artificial neural network.

GRANN is based on a standard statistical technique called kernel regression, which is similar to radial basis functions. With respect to an independent variable, x, and the training set, regression forecasts the value of the dependent variable, y,

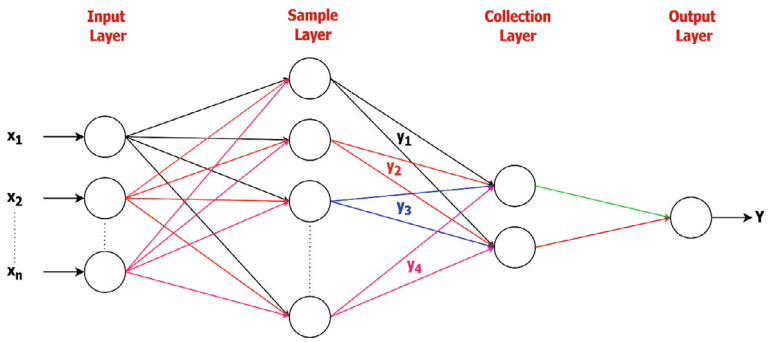


Fig. 9 Structure of GRANN

that is most likely to occur. The probability function (PF) of x and y in a training set can be estimated using the GRANN. Since PF is derived from data without the need for assumptions, the system is usually perfect [48, 49]. If the probability density function f(x, y) is known, the regression of the independent variable x with respect to the dependent variable y is obtained with the help of the equation specified in Eq. 2.

$$E\left[y|X\right] = \frac{\int_{-\infty}^{\infty} (X, y)dy}{\int_{-\infty}^{\infty} (X, y)dy} \tag{2}$$

Equations should be punctuated in the same way as ordinary text but with a small space before the end punctuation mark.

2.7 M5 Model Tree (M5-Tree)

The M5-Model Tree method was developed by Quinlan in 1992 [50]. This model, which is based on a two-component decision tree, is a new regression method. It describes the relationship between dependent and independent variables through a linear regression function applied to the last leaf nodes using the tree classification method. M5-Model tree used for both categorical and quantitative (numerical) data has better performance than other decision tree models. The M5-Model Tree structure consists of three types of nodes (root, branch, and leaves). Each node in the model represents a predictor variable. The purpose of using the splitting criterion in the formation of the model tree is to minimize the standard deviation of the nodes by explaining the error rate of the nodes. In cases where it is not possible to minimize the standard deviation of nodes, the node ends as a leaf. The reduction in standard deviation at a particular node is obtained with the help of the equation specified in Eq. 3 [51–55].

$$SDR = SD(T) - \sum \frac{|Ti|}{|T|} sd\,(Ti) \tag{3}$$

In this equation, T represents a set of training (sample) data acting on the node, sd represents the standard deviation. Ti refers to a subset of samples containing the i'th results of the potential data.

2.8 Multivariate Adaptive Regression Spline (MARS))

Developed by Stanford University statistician Jerome H. Friedman in 1991, the MARS model is a linear, nonparametric stepwise regression technique [56]. The MARS model emerged because of the combination of iterative separation and projective tracking regression methods. This model aims to predict non-linear continuous numerical results [57, 58].

In the MARS model, there is no assumption regarding the form of functional relationship between dependent and independent variables. The model creates a link between variables based on the basic functions obtained from its original data (regression data) and the coefficients associated with these basic functions. Thanks to this method, independent variable values (data space) are divided into different regions and a flexible regression model is established by associating each region with a regression equation. This makes the MARS model suitable and advantageous for solving multivariate regression problems.

Compared to traditional models, MARS has been shown to provide advantages in regression modeling processes by determining the complex data structure of multidimensional data, the interrelationships of the data, and providing optimal data transformation, which is not impossible but extremely difficult to solve. While developing a good regression model even for a model with a small data set requires a lot of effort and time, thanks to MARS, even large data sets and regression models with very complex data structures can be easily modeled and developed [59–62].

In the basic functions used in the Mars model, the x variable is produced in a piecewise and linear structure. The basic functions $(x\text{-}c)_+$ and $(c\text{-}x)_+$ are given below. Both functions are linear and piecewise. (+ sign represents the positive direction.) These functions;

$$(x-c)_+ = \begin{cases} x-c, & x > c \\ 0, & othercases \end{cases} \tag{4}$$

$$(c-x)_+ = \begin{cases} c-x, & x < c \\ 0, & othercases \end{cases} \tag{5}$$

Another equation of the basic functions (x-c) + and (c-x) + is $(\boldsymbol{x}-\boldsymbol{c})_+ = max(x-c, 0)$ ve $(c-x)_+ = max(c-x, 0)$ [60].

2.9 Support Vector Machines (SVM)

Vladimir Vapnik and Chervonenkis developed Support Vector Machines, a machine learning technique based on statistical learning theory, in 1992. [63]. Support vector regression (SVR) is the term used in the literature to describe regression that makes use of SVMs [64]. The goal of SVR is to identify a function that may roughly depict

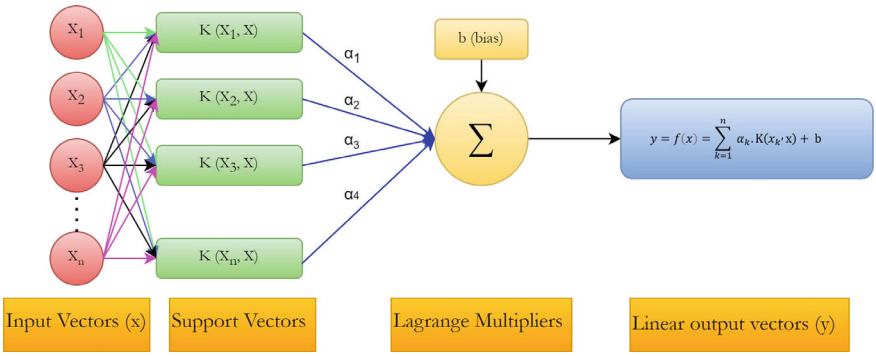


Fig. 10 Structure of support vector regression

the relationship between the input x and the output y(x). Figure 10 displays the architecture of SVR [65]:

The observed lake water level values are the input vectors (x) in this study, while the forecast values are the output values (y). The following is the form of the function that SVR attempts to forecast [66]:

$$y = f(x) = w.\phi(x) + b \tag{6}$$

In the equation, w is the weight vector, b is the deviation and ϕ is the transfer function.

3 Applications

In this section, the machine learning techniques used in this study and the comparison criteria of these techniques are mentioned. The best results obtained from the modelling results were compared. In this study, six different machine learning techniques were used. These were examined under the headings "*Artificial neural network methods*", "*Heuristic regression methods*" and "*Support vector machine*". Among these techniques, the methods that can be examined under the title of ANN methods are MLANN, GRANN and RBANN. Methods that can be examined under heuristic regression methods are M5-Tree (or M5-Model Tree) and MARS methods. For the support vector machine method, different kernel methods were investigated. For each technique, four data packages (P1, P2, P3, P4) with 14 inputs (1 to 14 months lag-time lake level) were modeled separately. Each data package consists of data packages with different combinations, 75% of which contains training data and 25% of which contains test data. The application of the study was carried out according to the material and method section and the flowchart of this study is given below Fig. 11.

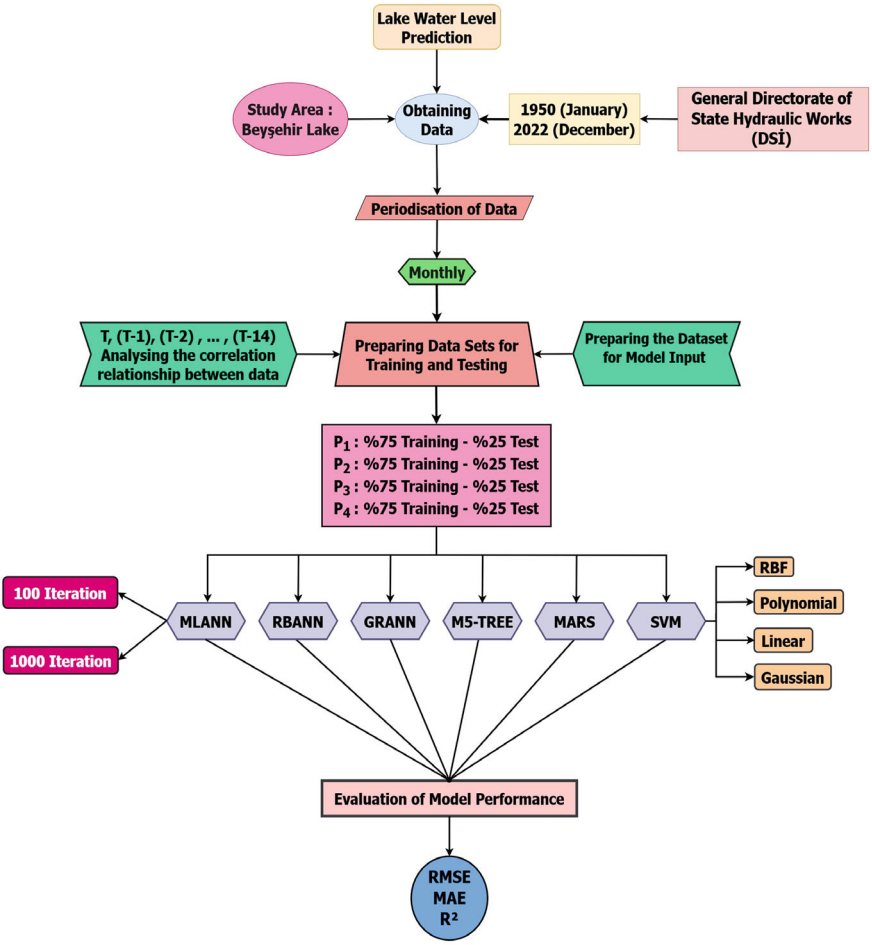


Fig. 11 Flowchart of the study

In modeling, two different iterations (100 iterations, 1000 iterations) were tried for multilayer ANNs to investigate whether the number of iterations affected the results. In addition, separate analyzes were carried out with gaussian, linear, polynomial and radial basis kernel functions for support vector machines and the effect of kernel functions on the analysis results was examined. Since there are no parameters for heuristic regression methods, the results were investigated only according to the number of inputs (1 to 14). The generally preferred criteria in the literature are used to evaluate models [67, 68]. In the analyses, comparison metrics such as RMSE, MAE, and R^2 were utilized [65]. Below are the formulas for each of the comparison criteria.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_e - Y_o)^2} \tag{7}$$

$$MAE = \frac{1}{N}\sum_{i==1}^{N}|Y_e - Y_o| \tag{8}$$

$$R^2 = \left(\frac{N\,x\left(\sum Y_o\,x\,Y_e\right) - \left(\sum Y_o\right)x\left(\sum Y_e\right)}{\sqrt{\left(N\,x\,\sum Y_o^2\right) - \left(\sum Y_o\right)^2 x\left(N\,x\,\sum Y_e^2\right) - \left(\sum Y_e\right)^2}}\right)^2 \tag{9}$$

In these equations, N represents the number of data, Y_e represents the model-predicted lake water level, and Y_o represents the observed lake water level data. The data used is the water level data of Lake Beyşehir. Therefore, the units are in meters (m). Since RMSE and MSE, which are among our comparison criteria, are also included in the error results, the error results are also in meters (m). High error results indicate that the model is far from real data (observed) and produces wrong (unsuccessful) results. Conversely, low results indicate that the model is close to the real data, has a low error rate, and is therefore suitable. The R^2 determination coefficient varies between 0 and 1, and higher values of the R^2 coefficient indicate greater compatibility and accuracy of the model with reality. The findings of the analyzes detailed in Figure 11 is presented in Table 3 (for ANNs), Table 4 (for heuristic regression methods) and Table 5 (SVMs).

Table 3 shows that the most successful method of three different ANN methods in P1, P2, P3 and P4 data sets is radial based ANNs. It is observed that the most successful result of RBANN is in input 12 of the P1 data set, with the best coefficient of determination and the lowest error value. In addition, it is understood that the P1 data set is the most successful data set for these three different artificial neural network methods, with its lower error value and higher coefficient of determination than the other data sets. As a result of the analyzes conduct with 100 and 1000 iterations of multilayer artificial neural networks, it was observed that increasing the number of iterations did not improve the analysis results. The order of the methods from most successful to least successful is RBANN, MLANN, GRANN.

In Table 4, two different heuristic regression methods are examined. As a result of the examinations, it was observed that the MARS method was the most successful method, and it achieved the most successful results in four different data packages. In Table 5, the methods of heuristic regression methods were examined in separate tables to better understand the separate analysis of four different kernel structures of SVM and the effect of kernel functions on the analysis results.

When the results of the SVM technique are examined in Table 5, it is observed that the most successful kernel method in the P1 and P3 data packages is the "linear" kernel method, while it is observed that the most successful kernel method in the P2 and P3 data packages is the "polynomial", "RBF" kernel, respectively. When the results of all data sets are compared, P1 data set is more successful in analysis results

Table 3 The best results of ANN methods according to data packets

Data Set	Model	Inputs	Training			Test		
			RMSE	MAE	R^2	RMSE	MAE	R^2
P_1	MLANN (100)	XII	0.164	0.105	0.977	0.118	0.084	0.976
	MLANN (1000)	XIII	0.160	0.102	0.978	0.118	0.083	0.976
	GRANN	II	0.142	0.096	0.983	0.157	0.110	0.956
	RBANN	XII	0.163	0.103	0.977	0.114	0.081	0.977
P_2	MLANN (100)	XIV	0.144	0.093	0.968	0.208	0.153	0.951
	MLANN (1000)	XIV	0.144	0.093	0.968	0.205	0.151	0.952
	GRANN	II	0.160	0.118	0.961	0.240	0.181	0.932
	RBANN	XII	0.143	0.091	0.969	0.184	0.126	0.953
P_3	MLANN (100)	XIII	0.141	0.089	0.980	0.165	0.100	0.940
	MLANN (1000)	XIII	0.141	0.089	0.980	0.165	0.100	0.940
	GRANN	I	0.189	0.141	0.963	0.201	0.143	0.911
	RBANN	XIV	0.154	0.102	0.976	0.160	0.102	0.944
P_4	MLANN (100)	XI	0.150	0.097	0.979	0.159	0.097	0.956
	MLANN (1000)	II	0.154	0.107	0.978	0.157	0.104	0.958
	GRANN	II	0.135	0.094	0.983	0.181	0.119	0.943
	RBANN	XI	0.154	0.099	0.978	0.153	0.095	0.959

Table 4 The best results of heuristic regression methods according to data packets

Data Set	Model	Inputs	Training			Test		
			RMSE	MAE	R^2	RMSE	MAE	R^2
P_1	M5-TREE	I	0.197	0.139	0.967	0.178	0.138	0.944
	MARS	XIII	0.157	0.097	0.979	0.118	0.084	0.975
P_2	M5-TREE	I	0.191	0.140	0.944	0.207	0.155	0.933
	MARS	XIV	0.139	0.088	0.970	0.194	0.118	0.944
P_3	M5-TREE	XIV	0.094	0.052	0.991	0.187	0.126	0.924
	MARS	XII	0.145	0.091	0.979	0.169	0.110	0.938
P_4	M5-TREE	VI	0.089	0.056	0.993	0.184	0.128	0.941
	MARS	XIII	0.146	0.093	0.980	0.154	0.093	0.959

Table 5 Best results of SVM methods according to data packages

Data Set	Model	Inputs	Training			Test		
			RMSE	MAE	R^2	RMSE	MAE	R^2
P_1	SVM-Gaussian	II	0.174	0.115	0.974	0.366	0.306	0.969
	SVM-Linear	II	0.174	0.115	0.974	0.366	0.306	0.969
	SVM-Polynomial	II	0.174	0.117	0.974	0.387	0.320	0.968
	SVM-RBF	II	0.174	0.115	0.974	0.366	0.306	0.969
P_2	SVM-Gaussian	I	0.192	0.141	0.944	1.415	1.392	0.895
	SVM-Linear	XIV	0.144	0.088	0.968	1.428	1.418	0.954
	SVM-Polynomial	VI	0.128	0.084	0.975	1.410	1.400	0.955
	SVM-RBF	I	0.192	0.141	0.944	1.415	1.392	0.895
P_3	SVM-Gaussian	I	0.193	0.143	0.962	1.008	0.944	0.906
	SVM-Linear	XIII	0.144	0.092	0.979	1.006	0.938	0.941
	SVM-Polynomial	II	0.155	0.106	0.976	1.014	0.943	0.933
	SVM-RBF	I	0.193	0.143	0.962	1.008	0.944	0.906
P_4	SVM-Gaussian	II	0.165	0.111	0.975	0.651	0.591	0.932
	SVM-Linear	XII	0.150	0.096	0.979	0.692	0.620	0.959
	SVM-Polynomial	II	0.165	0.112	0.975	0.709	0.635	0.953
	SVM-RBF	II	0.165	0.111	0.975	0.651	0.591	0.932

than other data sets. The most successful method (considering the RMSE value of the test phase [27]) and the best results for each data package are given in Table 6.

Table 6 Summary of best results

Data Set	Model	Inputs	Training			Test		
			RMSE	MAE	R^2	RMSE	MAE	R^2
P1	RBANN	XII	0.163	0.103	0.977	0.114	0.081	0.977
	MARS	XIII	0.157	0.097	0.979	0.118	0.084	0.975
	SVM-Linear	II	0.174	0.115	0.974	0.366	0.306	0.969
P2	RBANN	XII	0.143	0.091	0.969	0.184	0.126	0.953
	MARS	XIV	0.139	0.088	0.970	0.194	0.118	0.944
	SVM-Polynomial	VI	0.128	0.084	0.975	1.410	1.400	0.955
P3	RBANN	XIV	0.154	0.102	0.976	0.160	0.102	0.944
	MARS	XII	0.145	0.091	0.979	0.169	0.110	0.938
	SVM-Linear	XIII	0.144	0.092	0.979	1.006	0.938	0.941
P4	RBANN	XI	0.154	0.099	0.978	0.153	0.095	0.959
	MARS	XIII	0.146	0.093	0.980	0.154	0.093	0.959
	SVM-Linear	XII	0.150	0.096	0.979	0.692	0.620	0.959

When the best results of modeling examined with four different data sets, it is understood that the most successful method for data sets P1, P2, P3, and P4 is RBANN, RBANN method is followed by MARS method and lastly comes SVM. The fluctuations and distribution graphs in the observed and predicted lake level time series for these results are shown in Figs. 12, 13, 14 and 15.

Examining the graphs, it is possible to determine that our models are highly successful since they match at the peak and minimum points between the estimated values derived from our modeling and the actual observation values. Furthermore, successful approaches were typically obtained after a 12-month delay, based on the input data sets. In addition, Taylor diagram and Violin plot were used to graphically compare the best models. These diagrams provide a visual summary of how closely the models represent the data. Model correlations and mean square deviation were used to compare the models in the Taylor diagram [69]. In the Violin plot, models are

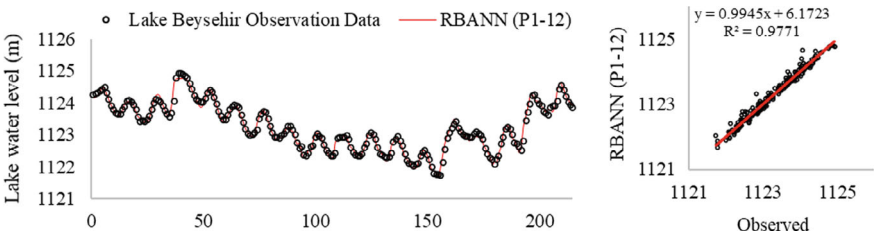


Fig. 12 Timeseries and scatter plots for P1 data set modelling

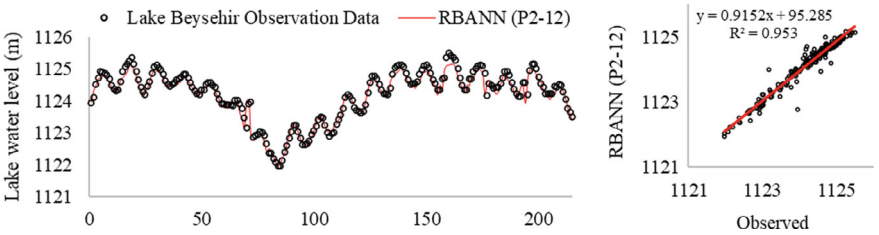


Fig. 13 Timeseries and scatter plots for P2 data set modelling

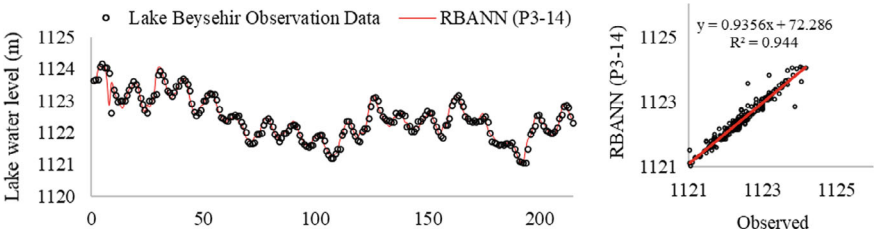


Fig. 14 Timeseries and scatter plots for P3 data set modelling

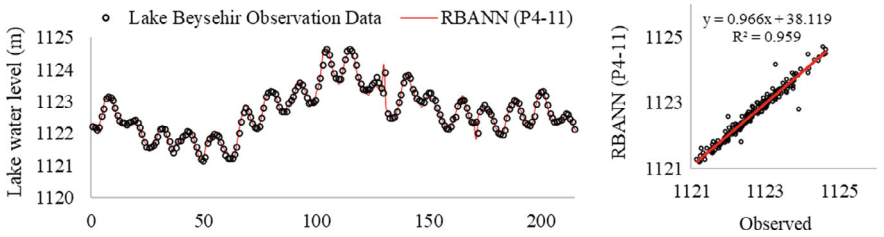


Fig. 15 Timeseries and scatter plots for P4 data set modelling

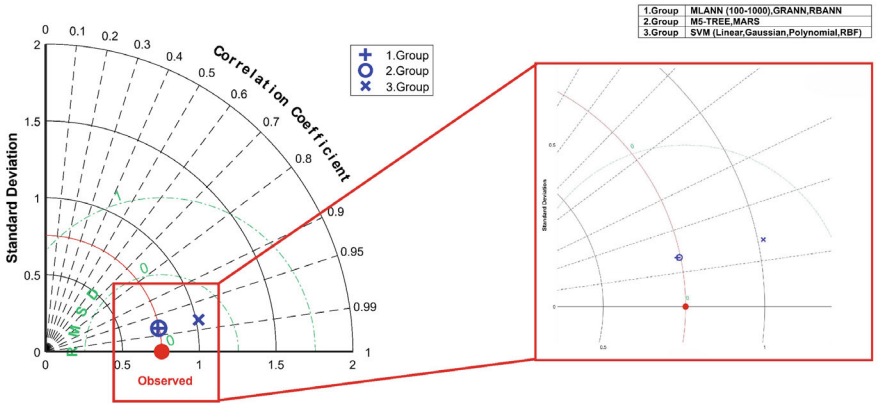


Fig. 16 Taylor diagram for P1 data set modelling

compared according to many statistical parameters such as mean, median standard deviation, etc. Taylor diagrams are shown below (Figs. 16, 17, 19 and 19) according to the best results of three different groups of models according to the test data sets (Pn).

When the closeness of the techniques in the Taylor diagrams to the observed data is examined, it is observed that the RBANN method has values close to the observation values in the charts of the P1, P2, P3 and P4 data sets. Violin plots are shown below (Figs. 20, 21, 22 and 23) according to the best results of 10 different models according to the test data sets (Pn).

The graphs of P1, P2, P3 and P4 data sets, it is seen that the RBANN method has values close to the observation Violin Chart Statistics. In addition, it has been observed that other methods generally comply with the observation data, except for the estimated values of SVM.

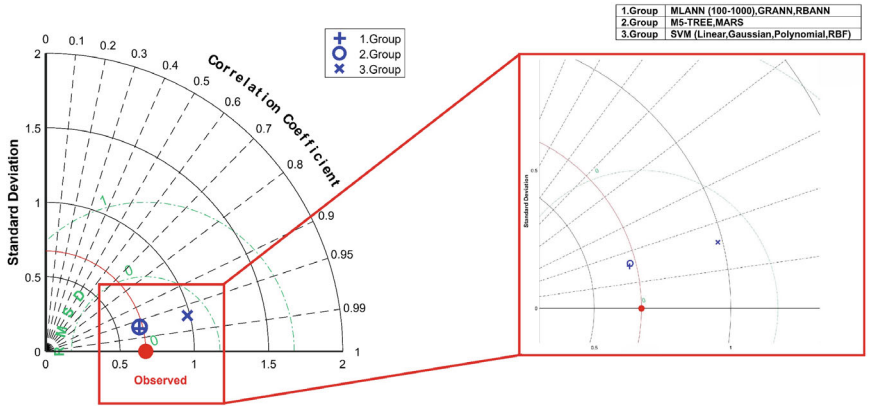


Fig. 17 Taylor diagram for P2 data set modelling

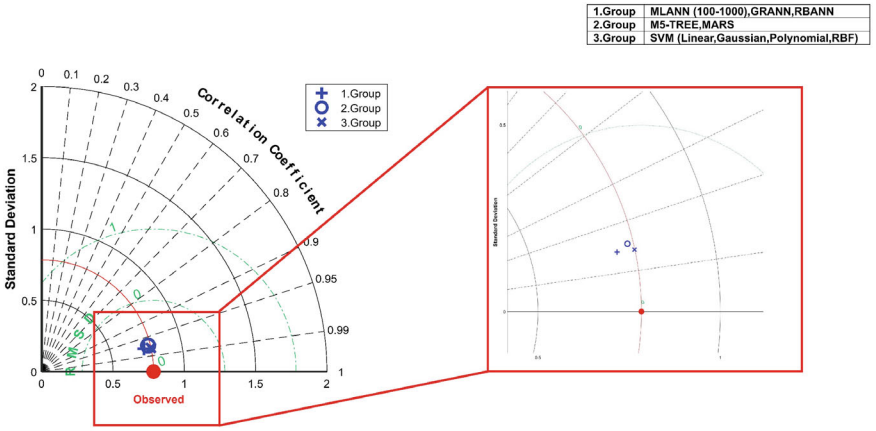


Fig. 18 Taylor diagram for P3 data set modelling

4 Conclusions

Predicting lake water levels, one of the topics addressed by civil engineers, ensures early warning in disaster management, thus ensuring the safety of individuals. It facilitates sustainable management of water resources, enhances agricultural productivity, supports environmental conservation efforts, and optimizes the operations of critical industries such as hydroelectric power generation. Machine learning techniques enable more accurate and effective predictions of lake water levels. In this study, predictions were obtained using six different techniques on various datasets by utilizing lake water level data from Lake Beyşehir, the largest freshwater lake in Turkey, spanning from 1950 to 2022.

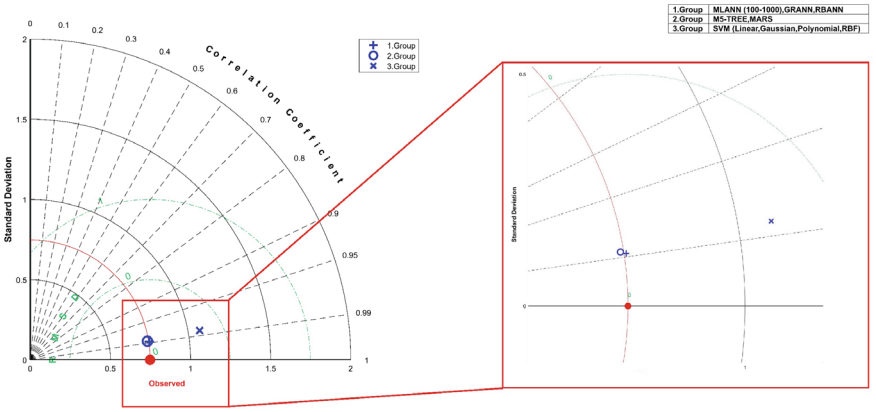


Fig. 19 Taylor diagram for P4 data set modelling

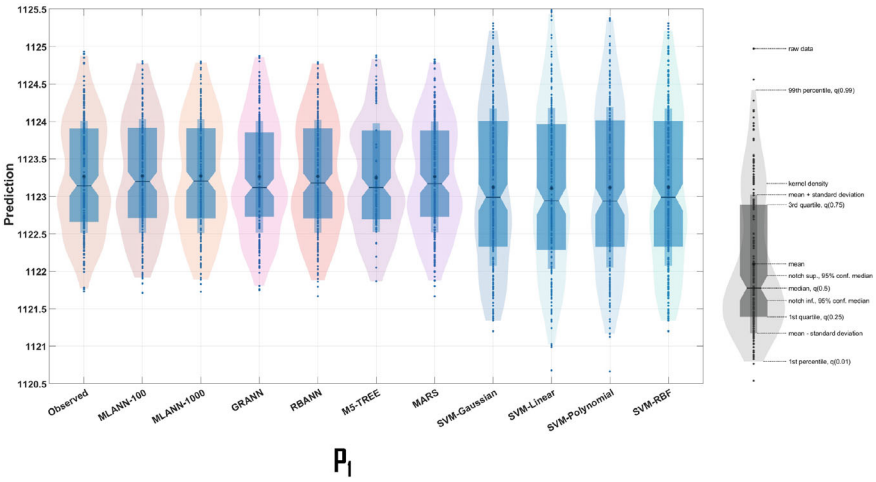


Fig. 20 Violin plot for P1 data set modelling

According to the analyses conducted with four different data sets for all methods, the method that achieved the most successful result is RBANN. The best result for this model is obtained from input 12 of data set P1 (RMSE = 0.114, MAE = 0.081, $R^2 = 0.977$). Additionally, for the other data sets (P2, P3, and P4), the best results obtained among all methods are as follows;

- For data set P2, the most successful result belongs to entry 12 of the RBANN method (RMSE = 0.184, MAE = 0.126, $R^2 = 0.953$).
- For data set P3, the most successful result belongs to entry 14 of the RBANN method (RMSE = 0.160, MAE = 0.102, $R^2 = 0.944$).

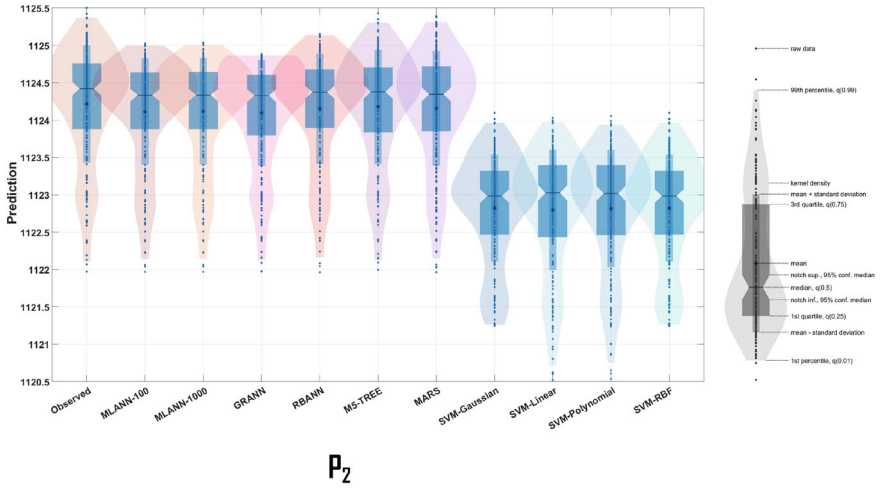


Fig. 21 Violin plot for P2 data set modelling

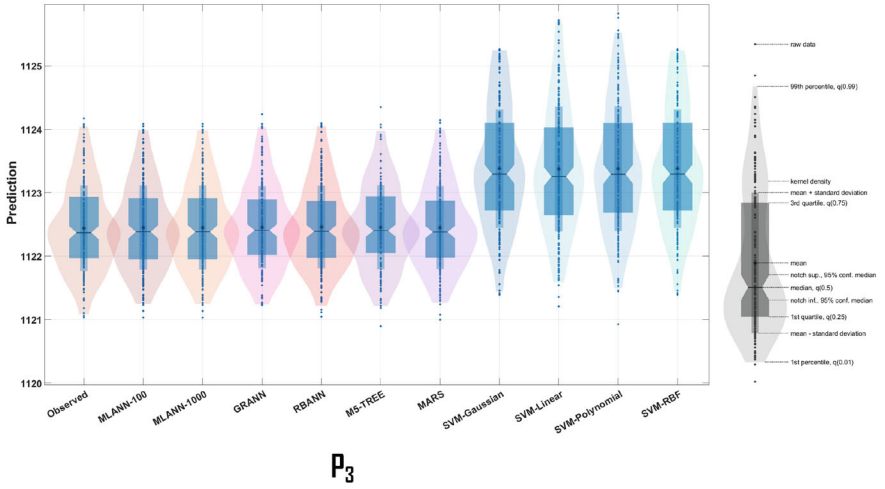


Fig. 22 Violin plot for P3 data set modelling

- For data set P4, the most successful result belongs to entry 11 of the RBANN method (RMSE = 0.153, MAE = 0.095, $R^2 = 0.959$).

This study observed that among the three different artificial neural network methods used, RBANN, MLANN, and GRANN yielded the most successful results for all data sets. Additionally, to determine the effect of the number of iterations on the results for multilayer artificial neural networks, analyses were conducted with 100 and 1000 iterations. It was observed that increasing the number of iterations did

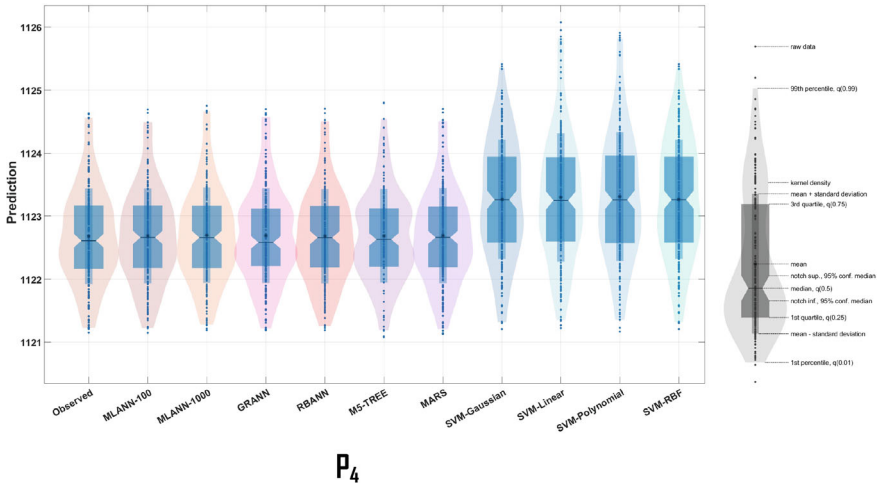


Fig. 23 Violin plot for P4 data set modelling

not improve the analysis results. Moreover, analyses performed with 1000 iterations took approximately 3.3 times longer than analyses conducted with 100 iterations. When the results of the M5-TREE and MARS methods for four different data sets were analyzed, it was observed that the MARS method achieved more successful results for each data set compared to the M5-Tree method. In the separate analyses of the P1, P2, P3, and P4 data sets using the SVM method with four different kernel functions (linear, polynomial, radial basis, gaussian), it can generally be said that the support vector machine method with a linear kernel function achieved successful results compared to SVMs with other kernel functions. Additionally, it was observed that SVM methods with Gaussian and radial basis kernel functions yielded quite close results in analyses conducted with all data sets.

This study demonstrates that, with the utilization of machine learning techniques in the field of water resources, which is one of the application areas of civil engineering, seemingly complex problems can be solved. It is understood that predictions and planning regarding the quantity of our vital water resources, crucial for the survival of living organisms, can be successfully conducted. Furthermore, it is believed that the different machine learning techniques used in the study will guide future projects and academic publications in this field by providing insights into data set formation and analysis techniques.

Acknowledgements The results presented in this chapter are partially based on the MSc thesis titled “Estimation of Beyşehir Lake Water Level Using Different Machine Learning Techniques, KTO Karatay University, Konya, Türkiye, 2024”, which is gratefully acknowledged. Additionally, the authors would like to thank the Konya State Hydraulic Works 4th Regional Directorate and its employees for their support in providing the data.

References

1. Mcgill: Hydraulic and Water Resources Engineering (2023). Available: mcgill.ca/civil/undergrad/areas/water
2. Behzadi, S., Jalilzadeh, A.: Introducing a novel digital elevation model using artificial neural network algorithm. Civil Eng. Dimens. **22**(2), 47–51 (2020). https://doi.org/10.9744/ced.22.2.47-51
3. Musie, W., Gonfa, G.: Fresh water resource, scarcity, water salinity challenges and possible remedies: a review. Heliyon **9**(8), 1–18 (2023). https://doi.org/10.1016/j.heliyon.2023.e18685
4. Hosseiny, S.H., Bozorg-Haddad, O., Bocchiola, D.: Water, culture, civilization, and history, Economical, Political, and Social Issues in Water Resources, pp. 189–216 (2021). https://doi.org/10.1016/B978-0-323-90567-1.00010-3
5. Wada, Y., van Beek, L.P.H., Viviroli, D., Dürr, H.H., Weingartner, R., Bierkens, M.F.P.: Global monthly water stress: 2. Water demand and severity of water stress. Water Resour. Res. **47**(7), 1–17 (2011). https://doi.org/10.1029/2010WR009792
6. Şen, Z., Kadioğlu, M., Batur, E.: Cluster regression model and level fluctuation features of Van Lake, Turkey. Ann. Geophys. **17**, 273–279 (1999). https://doi.org/10.1007/s00585-999-0273-4
7. Şen, Z., Kadioğlu, M., Batur, E.: Stochastic modeling of the Van Lake monthly level fluctuations in Turkey. Theor. Appl. Climatol. **65**, 99–110 (2000). https://doi.org/10.1007/s007040050007
8. Li, P., Wu, J.: Water resources and sustainable development. Water **16**(1), 134–140 (2023). https://doi.org/10.3390/w16010134
9. Yagbasan, O., Yazicigil, H., Demir, V.: Impacts of climatic variables on water-level variations in two shallow Eastern Mediterranean lakes. Environ. Earth Sci. **76**(16), 1–11 (2017). https://doi.org/10.1007/s12665-017-6917-x
10. Yagbasan, O., Demir, V., Yazicigil, H.: Trend analyses of meteorological variables and lake levels for two shallow lakes in central Turkey. Water **12**(2), 414–429 (2020). https://doi.org/10.3390/w12020414
11. Demir, A.: The effects of global climate change on biodiversity and ecosystems resources. Ankara Univ. J. Environ. Sci. **1**, 37–54 (2009). https://doi.org/10.1501/CSAUM_0000000013
12. Demir, V.: Trend analysis of lakes and sinkholes in the Konya Closed Basin, in Turkey. Nat. Hazards **112**, 2873–2912 (2022). https://doi.org/10.1007/s11069-022-05327-6
13. Leira, M., Cantonati, M.: Effects of water-level fluctuations on lakes: an annotated bibliography. Hydrobiologia **613**, 171–184 (2008). https://doi.org/10.1007/s10750-008-9465-2
14. Demir, V., Uray, E., Carbas, S.: Modeling Civil Engineering Problems via Hybrid Versions of Machine Learning and Metaheuristic Optimization Algorithms, Hybrid Metaheuristics in Structural Engineering, Studies in Systems, pp.199–234 (2023). https://doi.org/10.1007/978-3-031-34728-3
15. Zhu, S., Lu, H., Ptak, M., Dai, J., Ji, Q.: Lake water-level fluctuation forecasting using machine learning models: a systematic review. Environ. Sci. Pollut. Res. **27**, 44807–44819 (2020). https://doi.org/10.1007/s11356-020-10917-7
16. VOSviewer. Welcome to VOSviewer: Centre for Science and Technology Studies, Leiden University, The Netherlands (2022). Available: https://www.vosviewer.com/
17. van Eck, N.J., Waltman, L.: Software survey: VOSviewer, a computer program for bibliometric mapping. Scientometrics **84**, 523–538 (2010). https://doi.org/10.1007/s11192-009-0146-3
18. Altunkaynak, A.: Forecasting surface water level fluctuations of Lake Van by artificial neural networks. Water Resour. Manag. **21**, 399–408 (2007). https://doi.org/10.1007/s11269-006-9022-6
19. Yarar, A., Onüçyıldız, M.: Determination of water level fluctuations of beysehir lake using artificial neural networks. Selcuk. Univ. J. Eng. Sci. Technol. **24**(2), 21–30 (2009), Available: https://dergipark.org.tr/tr/pub/sujest/issue/23252/248206
20. Yarar, A., Onucyıldız, M., Copty, N.K.: Modelling level change in lakes using neuro-fuzzy and artificial neural networks. J. Hydrol. **365**, 329–334 (2009). https://doi.org/10.1016/j.jhydrol.2008.12.006

21. Karimi, S., Shiri, J., Kisi, O., Makarynskyy, O.: Forecasting water level fluctuations of Urmieh lake using gene expression programming and adaptive neuro-fuzzy inference system. Int. J. Ocean Clim. Syst. **3**, 109–125 (2012). https://doi.org/10.1260/1759-3131.3.2.109
22. Shiri, J., Shamshirband, S., Kisi, O., Karimi, S., Bateni, S.M., Hosseini Nezhad, S.H.: Prediction of water-level in the Urmia lake using the extreme learning machine approach. Water Resour. Manag. **30**, 5217–5229 (2016). https://doi.org/10.1007/s11269-016-1480-x
23. Zhu, S., Hrnjica, B., Ptak, M., Choiński, A., Sivakumar, B.: Forecasting of water level in multiple temperate lakes using machine learning models. J. Hydrol. **585**, 124819 (1–13) (2020). https://doi.org/10.1016/J.JHYDROL.2020.124819
24. Wang, Q., Wang, S.: Machine learning-based water level prediction in Lake erie, Water (Switzerland) **12**(10), 2654 (1–14) (2020). https://doi.org/10.3390/w12102654
25. Peprah, M.S., Larbi, E.K.: Lake water level prediction model based on artificial intelligence and classical techniques—an empirical study on Lake Volta Basin, Ghana. Int. J. Earth Sci. Knowl. Appl. **3**(2), 134–150 (2021). Available: https://dergipark.org.tr/en/pub/ijeska/issue/60684/896435
26. Çubukçu, E.A., Yılmaz, C.B., Demir, V., Sevimli, M.F.: Forecasting of monthly average lake levels of Lake Michigan with artificial neural networks, 1st Advanced Engineering Days (AED), pp. 4–7 (2021)
27. Demir, V., Yaseen, Z.M.: Neurocomputing intelligence models for lakes water level forecasting: a comprehensive review. Neural Comput. Appl. **35**(1), 303–343 (2022). https://doi.org/10.1007/s00521-022-07699-z
28. Ozdemir, S., Yaqub, M., Yildirim, S.O.: A systematic literature review on lake water level prediction models. Environ. Model Softw. **163**, 10568, 1–17 (2023), https://doi.org/10.1016/j.envsoft.2023.105684
29. Zakaria, M.N.A., Ahmed, A.N., Malek, M.A., Birima, A.H., Khan, M.M.H., Sherif, M., et al.: Exploring machine learning algorithms for accurate water level forecasting in Muda river, Malaysia. Heliyon **9**(7), 1–15 (2023). https://doi.org/10.1016/j.helıyon.2023.e17689
30. Muşmal, H.: Lake Beyşehir in the early 20th century and the 1910–1911 Great Floods, Selçuk Univ. J. Turkic Stud., pp. 219–262 (2008), Available: https://dergipark.org.tr/tr/pub/sutad/issue/26267/276724
31. Parklar M. Beyşehir Lake National Park. Available: https://bolge8.tarimorman.gov.tr/menu/19/milli-parklar
32. Soyaslan, İ, Hepdeniz, K.: Evaluation of hydrogeological features based on groundwater flow modeling in the Beyşehir Lake Basin. J. Grad Sch. Nat. Appl. Sci. Mehmet Akif Ersoy Univ. **9**, 63–74 (2018). https://doi.org/10.29048/makufebed.358206
33. Doğan, A., Şanli, A.S.: Optimization of Beyşehir Lake water use based on climate data. Dicle Univ. Fac. Eng. Eng. J. **7**, 191–199 (2016), Available: https://dergipark.org.tr/tr/pub/dumf/issue/29220/312757
34. Elmas, Ç.: Artificial Intelligence Applications, 5th ed, Seçkin Publishing. (In Turkish) (2021)
35. Kubat, C.: Matlab—Artificial Intelligence and Engineering Applications, 3rd ed, Abaküs Publishing (2017). (In Turkish)
36. Öztemel, E.: Artificial Neural Networks, 3rd ed, Papatya Publishing (2020). (In Turkish)
37. Gültepe, Y.A.: Comparative assessment on air pollution estimation by machine learning algorithms. Eur. J. Sci. Technol. 16, 8–15 (2019). https://doi.org/10.31590/ejosat.530347
38. Yavuz, S., Deveci, M.: The effect of statistical normalization techniques on the performance of artificial neural network. J. Erciyes Univ. Fac. Econ. Adm. Sci. 40, 167–187 (2015), Available: https://dergipark.org.tr/tr/pub/erciyesiibd/issue/5897/78019
39. Yu, H., Wilamowski, B.M.: Levenberg–Marquardt Training, Industrial Electronics Handbook, 2nd ed., pp. 12/1–12/15. CRC Press (2011)
40. Altun, H., Eminoğlu, U., Tezekici, B.S.: Dependency of the learning process. In: MLP Artificial Neural Networks on The Activation Function And Input Data Showing Statistical Change, Eleco' 2002 Electrical Engineering Symposium, pp. 310–314 (2002), Available: http://www.emo.org.tr/ekler/490c742cd8318b8_ek.pdf

41. Kargı, V.S.A.: Artificial Neural Network Models and An Application at A Textile Firm, Uludağ University, Institute of Social Sciences, Ph.D. thesis (2013)
42. Keskenler, M.F., Keskenler, E.F.: From past to present artificial neural networks and history. Tak Vekayi **5**(2,8), 8–18 (2017), Available: https://dergipark.org.tr/tr/pub/takvim/issue/33375/346279
43. Okkan, U., Dalkiliç, H.Y.: Monthly runoff model for Kemer dam with radial based artificial neural networks. Tech. J. **23**(112), 5957–5966 (2012), Available: https://dergipark.org.tr/tr/pub/tekderg/issue/12747/155167
44. Partal, T., Kahya, E., Cığızoğlu, K.: Estimation of precipitation data using artificial neural networks and wavelet transform. ITU Eng. J. **7**, 73–85 (2008)
45. Şenkal, O.: Land surface temperature modeling for turkey using artificial neural networks. Sci. Eng. J. Fırat. Univ. **28**, 143–147 (2016), Available: https://dergipark.org.tr/tr/pub/fumbd/issue/29391/314609
46. Oral, M., Kartal, S., Özyıldırım, B.M.: A cluster based approach to reduce pattern layer size for generalized regression neural network. Pamukkale Univ. J. Eng. Sci. **24**(5), 857–863 (2018). https://doi.org/10.5505/pajes.2017.76401
47. Specht, D.F.: A general regression neural network. IEEE Trans. Neural Networks **2**(6), 568–576 (1991). https://doi.org/10.1109/72.97934
48. Sürel, A.: Use of Generalized Regression Artificial Neural Network in Water Resources Engineering, Master's Thesis, Istanbul Technical University, Institute of Science and Technology, Istanbul (2006)
49. Alp, M., Cığızoğlu, H.K.: Modelling rainfall-runoff relation using different artificial neural network methods. ITU Eng J. **3**(1), 80–88 (2004)
50. Quinlan, J.R.: Learning with continuous classes. World Sci. **92**, 343–348 (1992). 10.1.1.34.885
51. Bayatvarkeshi, M., Imteaz, M.A., Kisi, O., Zarei, M., Yaseen, Z.M.: Application of M5 model tree optimized with excel solver platform for water quality parameter estimation. Environ. Sci. Pollut. Res. **28**, 7347–7364 (2021). https://doi.org/10.1007/s11356-020-11047-w
52. Pal, M., Deswal, S.: M5 model tree based modelling of reference evapotranspiration. Hydrol. Process. **23**(10), 1437–1443 (2009). https://doi.org/10.1002/hyp.7266
53. Demir, V., Çubukçu, E.A.: Digital elevation modeling with heuristic regression techniques. Eur. J. Sci. Technol. **24**, 484–488 (2021), https://doi.org/10.31590/ejosat.916012
54. Kisi, O., Parmar, K.S., Soni, K., Demir, V.: Modeling of air pollutants using least square support vector regression, multivariate adaptive regression spline, and M5 model tree models. Air Qual. Atmos. Heal **10**, 873–883 (2017). https://doi.org/10.1007/s11869-017-0477-9
55. Demirgül, T., Demir, V., Sevimli, M.F.: Forecasting of HELIOSAT-based solar radiation by Model-Tree (M5-tree) approach. Geomatic **8**(2), 124–135 (2023)
56. Friedman, J.H.: Multivariate adaptive regression splines. Ann. Stat. **19**(1), 590–606 (1991), https://doi.org/10.1214/aos/1176347963
57. Demir, V., Yaseen, Z.M.: Neurocomputing intelligence models for lakes water level forecasting: a comprehensive review. Neural Comput. Appl. **35**, 303–343 (2023). https://doi.org/10.1007/s00521-022-07699-z/tables/15
58. Özfalcı, Y.: Multivariate adaptive regression splines: MARS, Master's Thesis, Gazi University, Art & Science Institute, Ankara (2008)
59. Briand, L.C., Freimut, B., Vollei, F.: Using multiple adaptive regression splines to support decision making in code inspections. J. Syst. Softw. **73**(2), 205–217 (2004). https://doi.org/10.1016/J.JSS.2004.01.015
60. Gülcüoğlu, E.: Investigation of secondary school student performance with MARS model, Master's Thesis, Bartın University, Postgraduate Education Institute, Bartın (2022)
61. Nacar, S., Kankal, M., Hınıs, M.A.: Estimation of daily streamflow using multivariate adaptive regression splines (MARS)-a case study of Haldizen stream. Gumushane Univ. J. Sci. Technol. **8**(1), 38–47 (2018). https://doi.org/10.17714/gumusfenbıl.311188
62. Toprak, S.: Time series modelling using multivariate adaptive regression splines and conic quadratic programming, Master's Thesis, Dicle University, Institute of Natural and Applied Sciences, Diyarbakır (2011)

63. Vapnik, V.: The support vector method of function estimation. Nonlinear Modeling, pp. 55–85, Springer (1998)
64. Smola, A., Burges, C., Drucker, H., Golowich, S., van Hemmen, L., Muller, K.R., et al.: Regression Estimation with Support Vector Learning Machines in collaboration with, 1–79 (1996)
65. Tao, H., Abba, S.I., Al-Areeq, A.M., Tangang, F., Samantaray, S., Sahoo, A., et al.: Hybridized artificial intelligence models with nature-inspired algorithms for river flow modeling: A comprehensive review, assessment, and possible future research directions. Eng. Appl. Artif. Intell. **129**(107559), 1–46 (2024), https://doi.org/10.1016/j.engappai.2023.107559
66. Abba, S.I., Kılınc, H.C., Tan, M.L., Demir, V., Ahmadianfar, I., Halder, B. et al.: Bio-communal wastewater treatment plant real-time modeling using an intelligent meta-heuristic approach: a sustainable and green ecosystem. J. Water Process. Eng. **53**, 1–18 (103731) (2023), https://doi.org/10.1016/j.jwpe.2023.103731
67. Legates, D.R., McCabe, G.J., Jr.: Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water Resour. Res. **35**(1), 233–241 (1999)
68. Heddam, S., Al-Areeq, A.M., Tan, M.L., Ahmadianfar, I., Halder, B., Demir, V., et al.: New formulation for predicting total dissolved gas supersaturation in dam reservoir: application of hybrid artificial intelligence models based on multiple signal decomposition. Artif. Intell. Rev. **57**, 1–57 (2024). https://doi.org/10.1007/s10462-024-10707-4
69. Demir, V., Citakoglu, H.: Forecasting of solar radiation using different machine learning approaches. Neural Comput. Appl. **35**(1), 887–906 (2023). https://doi.org/10.1007/s00521-022-07841-x

Artificial Intelligence-Driven Structural Health Monitoring: Challenges, Progress, and Applications

Victor Higino Meneguitte Alves, Vinicius Antônio Meneguitte Alves, and Alexandre Abrahão Cury

Abstract This chapter provides a comprehensive study and critical reflections on the use of artificial intelligence techniques to detect structural deterioration using vibration signals (such as accelerations, displacements, and so on). Machine learning and deep learning-based approaches are seen as promising tools for improving safety and optimizing preventive maintenance plans. However, some authors recognize concerns arising from strictly supervised methods, the "black box" nature of the models and their interpretability by human operators. As a result, the purpose of this work is to give useful information about the current damage detection paradigm, allowing for real-time, non-destructive, and trustworthy predictions regarding construction safety within the context of Industry 4.0. Moreover, issues associated with the application of artificial intelligence for pattern identification and decision-making in monitoring structural anomalies are highlighted and evaluated in recent studies.

Keywords Damage detection · Buildings · Computational intelligence · Machine learning · Deep learning

List of Abbreviations

AE	Auto-Encoders
AI	Artificial Intelligence
ANN	Artificial Neural Networks

V. H. M. Alves · V. A. M. Alves · A. A. Cury (✉)
Graduate Program in Civil Engineering, Faculty of Engineering, University of Juiz de Fora, São Pedro, Juiz de Fora, Minas Gerais 36036-900, Brazil
e-mail: alexandre.cury@ufjf.br

V. H. M. Alves
e-mail: victor.meneguitte@engenharia.ufjf.br

V. A. M. Alves
e-mail: vinicius.meneguitte@engenharia.ufjf.br

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_7

AR	AutoRegressive
BPNN	Back Propagation Neural Network
CNC	Computer Numerical Control
CNN	Convolutional Neural Network
CycleGAN	Cycle Generative Adversarial Network
DL	Deep Learning
DRN	Deep Residual Network
DS	Data Science
DT	Decision Trees
FE	Finite Element
FEM	Finite Element Method
FS	Feature Selection
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GS	Grid Search
KNN	K-Nearest Neighbour
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long Short-Term Memory
ML	Machine Learning
PSO	Particle Swarm Optimization
QUGS	Qatar University Grandstand Simulator
SAE	Sparse Auto-Encoder
SHAP	SHapley Additive exPlanations
SHM	Structural Health Monitoring
SVM	Support Vector Machines
TL	Transfer Learning
VSHM	Vibration Structural Health
WDCGAN	Wasserstein Deep Convolutional Generative Adversarial
WPRE	Wavelet Packet Relative Energy

1 Introduction

Predicting the current safety state of a particular structure is a crucial task for maintaining its performance throughout its lifespan. However, such prediction heavily relies on environmental conditions, usage loads, design specifications, and many other factors, resulting in a highly complex and often impractical interaction to be analyze altogether. Initially, structural assessment relies on human visual inspections, which can be unfeasible in remote access locations. Moreover, they represent a subjective analysis by the engineer/technician and are biased by their limitation to the exterior of the structural element. Internal damage to the structure cannot be usually identified through this method.

Therefore, Structural Health Monitoring (SHM) systems have gained significant attention from researchers across various domains such as Civil [1], Mechanics [2], Railway [3], and Aerospace [4] Engineering in recent years. This is due to their non-destructive and real-time approach, which holds the potential to save maintenance costs, as repair intervals can be optimized. Typically, SHM tasks are divided into four stages, with higher levels demanding more effort to be accomplished, namely: (I) detecting the presence of damage, (II) locating the damage, (III) quantifying the damage, and (IV) determining the remaining structure's lifespan [5]. The integration of these levels forms a robust framework supporting the SHM scientific community, providing them with a powerful tool to manage the complexities of damage identification.

In this context, most conventional methods for SHM rely on modal analysis, which involves identifying modal parameters, such as natural frequencies, damping ratios, and vibration modes [6]. However, this modal identification process acts as a filter that may lead to information loss, concealing subtle structural changes that could be associated with damage [7]. Indeed, research conducted by Ortiz Morales and Cury [7] has shown that, in general, it is not possible to establish a clear relationship between temperature variations and the effects of structural damage on the natural vibration frequencies of a simply supported beam tested in laboratory. Additionally, methods based on modal parameters assume that the structure remains within the linear elastic domain after the occurrence of damage, which does not always correspond to reality [8].

Another well-established approach, originating from computer vision, is thoroughly discussed by Dong and Catbas [9]. That method automates the visual inspection process but suffers from the limitation of only examining the exterior of structural elements. In their study, the authors highlight several challenges in implementing computer vision-based methods, such as shadow, uneven lighting, changes in camera tilt angle, and camera vibration. Consequently, anomalies or degradation processes may occur within structural components and remain masked from surface-level observation. Discontinuities, cracks, or voids in concrete may be among these underlying issues [10], as well as hidden defects, including embedded reinforcement corrosion, fractured fibers, or delamination cracks in composite materials [11]. On the other hand, vibration-based monitoring assesses structural integrity through dynamic data collected by sensors positioned on the structures themselves. The latter strategy is the primary focus in this chapter and proves to be extremely beneficial when visually identifying structural problems becomes difficult or impractical.

Recently, many approaches have been developed to harness the potential of utilizing artificial intelligence (AI) techniques in structural degradation detection. Among these tools, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Trees (DT), Convolutional Neural Networks (CNN), among others, stand out [12]. A standard damage detection system is depicted in Fig. 1, comprising two main elements: hardware and software. In the hardware part (Fig. 1b), sensors (i.e., accelerometers) distributed throughout the structure (Fig. 1a) are used to collect dynamic responses through an acquisition system, with these signals being generated by either ambient or forced excitation. Meanwhile, the software part (Fig. 1c) is

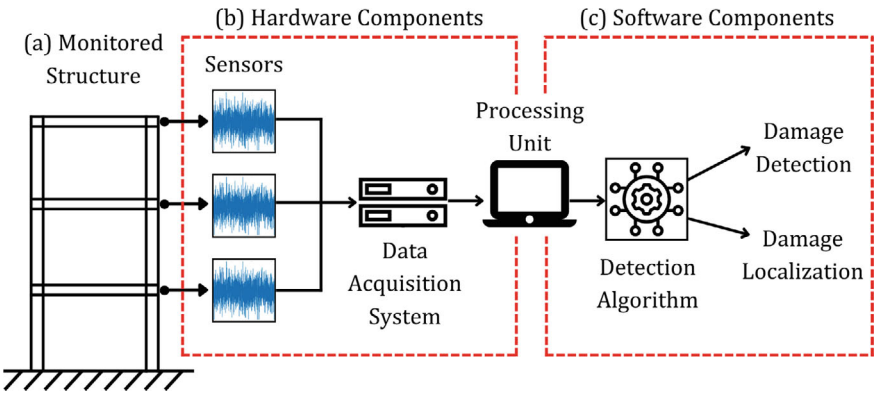


Fig. 1 Key components of structural damage identification systems

responsible for preprocessing the data, as well as post-processing that may involve a defined algorithm for pattern recognition, conducting damage detection, localization, and/or quantification. It is worth noting that nothing prevents the simultaneous execution of all four tasks (I–IV) by the same method. However, this problem becomes progressively trickier. Thus, it is common for some authors to limit themselves to investigating a specific task, such as "only" verifying the existence of damage (equivalent to level I), for example.

Current advances in sensor technology aim to enhance reliability and provide increasingly affordable options for large-scale applications. This represents a significant achievement as the accuracy of localization improves with the deployment of more sensors across the structure. Additionally, progresses in chip technology and processing power have enabled the integration of Industry 4.0 principles into condition monitoring practices. As a result, signal processing is an ever-expanding field and serves as a fundamental tool for SHM strategies, thanks to recent advances in Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL).

From this angle, this chapter attempts to provide a thorough overview of the problems and developments in structural monitoring using dynamic data, emphasizing the achievements made in this field in the larger picture of Industry 4.0. That is why, in contrast to other reviews, the present systematic evaluation compares the state-of-the-art SHM approaches grounded in data science, machine learning, and deep learning in terms of how well they execute tasks according to Rytter's scale [5]. Another relevant and original aspect of this work is the critical discussion of both supervised and unsupervised nature of the algorithms, its implications in real monitoring situations, and the methodologies that various authors have introduced to address these issues. Furthermore, this study surveys major available benchmark datasets, i.e., structures experimentally tested, which are recurrent in the literature as well-consolidated case studies with data open to the scientific community. Thus, the recommended database list facilitates comparison among state-of-the-art methods. Finally, ethical reflections on the interpretability of computational intelligence models, which tend to act

as "black boxes," are presented, followed by a discussion on available techniques to mitigate this type of behavior. Consequently, this chapter will also serve as an introductory manual of best practices within the field of AI-based SHM studies.

2 Methodology

The present chapter aims to provide updated bibliographic research related to SHM using ML/DL methods. Only papers published between 2017 and 2023 were considered to encompass the most current studies on the subject. It is worth noting that this compilation does not aim to provide an exhaustive overview of all SHM methods utilizing dynamic data developed to date, but rather a carefully selected collection of peer-reviewed articles, offering a panoramic view of the current state-of-the-art. Employing a systematic search strategy, the most relevant works were identified and included in this compilation, sourced from renowned academic databases such as Web of Science, Science Direct, Wiley Online Library, and Google Scholar, focusing on articles from prominent journals in the field of SHM studies. The goal is to provide a comprehensive analysis of the progress made in this field, acknowledging the advances and challenges that guide structural health monitoring through vibration data in the era of Industry 4.0. Boolean terms and connectors used for combining them in the search were: ("Vibration") AND ("Damage") AND ("Detection" OR "Localization") AND ("SHM") AND ("AI" OR "ML" OR "DL").

Subsequently, the titles, abstracts, and keywords were evaluated to exclude works unrelated to approaches involving the use of computational intelligence algorithms applied to structural health monitoring. From the remaining documents, 16 were selected, read in full, and judged by the authors to be the studies that best comprehensively represent the state-of-the-art on that topic. After reading, these 16 articles had their data cataloged, and their findings and results were presented and discussed. Additionally, this work is also based on literature review articles aiming to identify the most innovative characteristics in the field of structural damage detection using vibration data with AI. Therefore, for further details on the topic, the works of Avci et al. [12], Azimi et al. [13] are suggested. An overview of the methodology used in this study is illustrated in Fig. 2.

3 Anomaly Identification Techniques and Results

In this section, recent advances in structural damage detection research utilizing vibration data are discussed. Various methodologies involving data science, machine learning, and deep learning are investigated. Moreover, this exploration not only

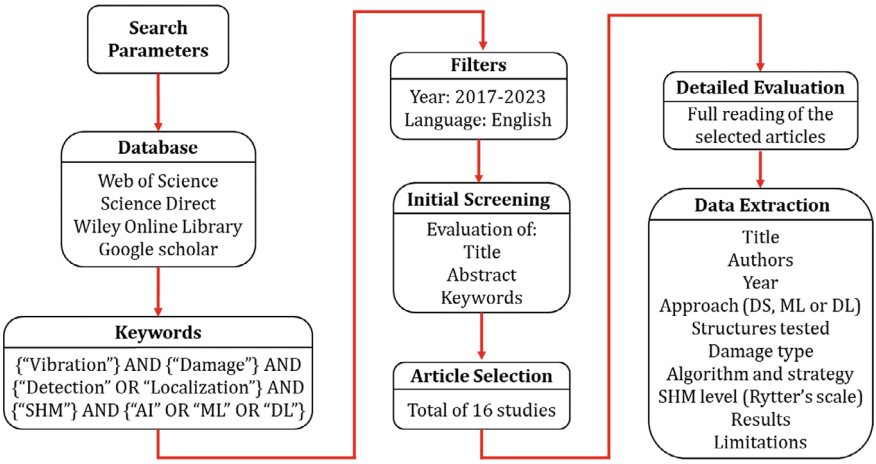


Fig. 2 Flowchart of the proposed systematic review

showcases the potential of these approaches but also sheds light on the inherent challenges and constraints associated with their implementation. Through this comprehensive exploration, we intend to provide insights that contribute to the ongoing evolution of SHM techniques.

3.1 Damage Detection Using Data Science (DS) and Machine Learning (ML)

The work by Alves and Cury [14] fundamentally employs concepts from data science and statistics to localize structural damage. By utilizing vibration data collected from both intact (reference) and current (unknown) states of the structure, features are extracted in the time, frequency, and quefrency domains. After extracting 35 features from the raw acceleration signals, it is important to construct an index that objectively reflects the location(s) of damage. Consequently, the suggested technique begins by determining percentile intervals for each of the characteristics associated with the intact state of the structure. Subsequently, an outlier analysis is conducted by comparing the characteristics of the unknown state to the intervals derived from the reference (healthy) structure. As a result, a damage index is calculated such that the higher the number of discrepant features at a particular location (sensor), the higher the damage index corresponding to a greater likelihood of damage. It is noticeable that this process of comparing intact signals with unknown signals from the possibly damaged structure is a recurring theme among other methodologies across the literature. Figure 3 illustrates this usual configuration.

The aforementioned methodology has been enhanced and fully automated in Alves and Cury [15]. However, this time, the extracted features were subsequently

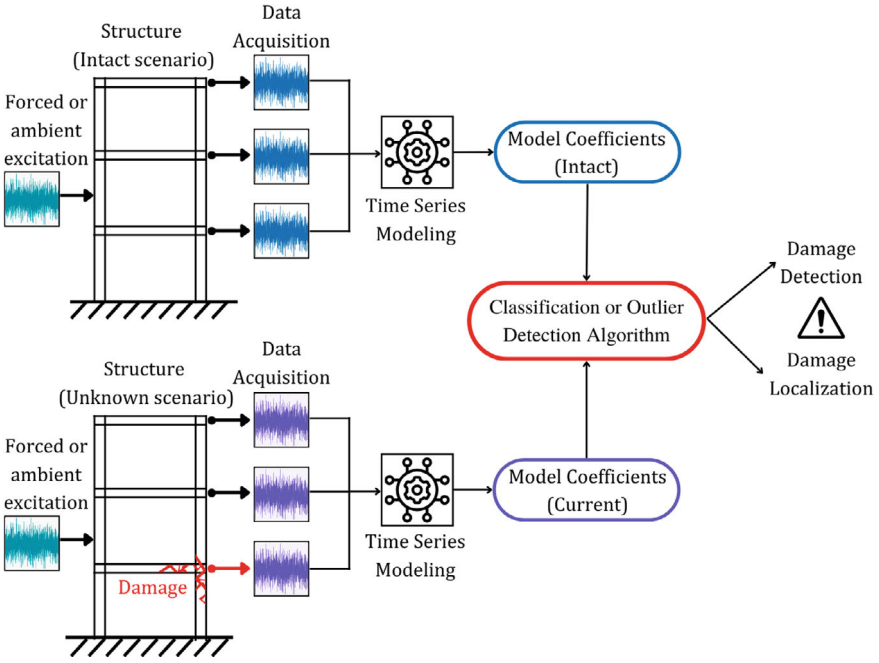


Fig. 3 Typical scheme of damage detection using artificial intelligence from dynamic signals

filtered by an unsupervised feature selection (FS) technique called InfFS, based on their relevance and non-redundancy. Moreover, a new weighted damage index was employed to examine the degree of importance within the selected feature set. Despite being entirely unsupervised and automated, the method achieved results comparable to traditional approaches and even recent deep learning techniques in all tested applications. In an experiment with a simply supported beam, the method proved to be promising in detecting extremely subtle damage, such as small holes, which exhibited variations of less than 1% of the first natural frequency. This capability demonstrates its feasibility in early detection of damage in scenarios where modal analysis-based approaches would be ineffective. Furthermore, the method equally showed its sensitivity in detecting considerably severe damage such as a 51.7% stiffness loss in 4 steel columns of a small-scale 2-D frame.

For both situations, only intact data was used to develop the detection model, i.e., an unsupervised procedure. However, there is a broad range of instances where supervised training schemes are employed. This raises the question of how damaged data would be generated to train the models, considering that intentionally damaging the structure would be contradictory.

A supervised strategy from the study by Gui et al. [16] investigated the viability of employing Support Vector Machines (SVM) coupled with optimization techniques for enhanced feature extraction and optimization parameters for damage detection. Two types of feature extraction methods were evaluated, including AutoRegressive

(AR) model parameters and statistical parameters residual errors. Subsequently, three optimization techniques were tested: Grid Search (GS), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA), to effectively determine the parameters in the SVM.

Therefore, it becomes clear that there is a debate over whether the features that are utilized as input have substantial different degrees of importance and how to handle it. From this perspective, the work of Ghiasi et al. [17] conducted feature extraction using wavelet packet relative energy (WPRE) and proposed a FS method to choose the most relevant and non-redundant features. An enhanced version of SVM was employed for classification. At each step, comparisons with other metrics were conducted, demonstrating higher accuracy in detecting the presence of damage. The method was tested on six structures, achieving accuracies ranging from 79.8 to 99.7%.

To mitigate the influence of external environmental factors, the study by Finotti et al. [18] utilized natural frequencies and temperature measurements of the structure as inputs for the SVM model. Experimental data from the Gabbia Tower in Italy were used to verify the proposed methodology, including records of structural behavior before, during, and after a seismic event. After analyzing the results, it was found that the technique based on natural frequencies was efficient in detecting structural changes, yielding results consistent with the actual situation of the tower, and allowing for clear identification of the damage caused by the seismic event. However, it highlights the challenge of acquiring training data for the models in different structural scenarios.

To address the issue of damaged data for training, from a model-based viewpoint, Mariniello et al. [19] generated a library of "pathological" and "non-pathological" virtual structural responses using finite element (FE) models calibrated by laboratory tests. They applied this library to a vibration methodology coupled with a decision tree-based (DT) learning method. The approach aimed to identify abnormal responses, indicating the location and severity of damage. While the method successfully located single damage, it exhibited reduced accuracy when dealing with two or more damaged elements. Overall, the proposed approach classified damage by learning the properties of a simulated dataset, highlighting the disadvantage that if the modeling of the structural system is not realistic enough, the approach becomes invalid.

To minimize these limitations, Zacharakis and Giagopoulos [20] introduced a damage detection technique using the Finite Element Method (FEM), meta-heuristic algorithms, and experimental acceleration data. The primary goal was to facilitate a vibration-based structural inspection strategy by leveraging monitoring information. As a starting point, they developed an FE model using a Particle Swarm Optimization (PSO) algorithm, capable of accurately describing the dynamic response of the healthy structure. The optimized FE model, along with experimental measurements of the structure, served as inputs for damage detection, while the output was a damaged FE model approximating the dynamic behavior of the real-world damaged structure. To achieve this, a damaged area was introduced into the optimized FE model by altering stiffness and mass parameters to approximate the effect of physical

damage. After the optimization procedure converged, the inserted area highlighted the damaged region of the structure. Two structures were tested: a vehicle chassis and a composite material bar. The maximum errors in modeling for the two tested structures were 2.38% and 2.14%, respectively. Additionally, damage scenarios induced changes in the dynamic response of the physical structures ranging from 2.5 to 5.39%. The proposed approach successfully identified the damaged area in all four scenarios studied.

Based on a comprehensive compilation of 22 case studies of data science in SHM, the book by Cury et al. [21] highlights several critical factors that impact the performance of damage identification algorithms in large-scale structural systems. These challenges are also evident in other studies presented in this section. Firstly, within the field of Big Data, the high dimensionality of monitored parameters poses a challenge in the effective analysis and processing of the vast amount of data involved. In addition, environmental and operational factors such as temperature, humidity, and traffic can introduce variability and uncertainty into measurements, further complicating the identification process. The inherent complexity of large structures also contributes to the difficulty in accurately detecting and locating damage. Moreover, ensuring the reliability of measured data is crucial for obtaining precise and trustworthy results. Another significant issue is the limited sensitivity of global structural responses to local damage, which can hinder the early detection of subtle anomalies. Finally, it is understood that effective structural integrity monitoring requires the integration of physical and engineering knowledge into machine learning algorithms to enable a data-driven approach for efficient decision-making.

3.2 Damage Detection via Deep Learning (DL)

Deep learning (DL) is being used in recent literature research to identify structural defects; this challenge is formulated as a time series classification assignment [22]. According to this standpoint, convolutional neural networks, or CNNs, are extensively employed as they represent the most advanced deep learning method available for identifying patterns in pictures [23]. However, since we are working with vectors (signals) rather than matrices (images), one-dimensional convolutional neural networks (1D CNNs) are frequently used for vibration signal applications. The advantage of this approach is its ability to automatically extract damage-sensitive features from acceleration signals through the training of subsequent layers of neurons, while other conventional data mining methods use hand-picked features, which may be suboptimal for a particular structure or fail to achieve the same level of performance across different structures.

Hence, the strategy developed in the work of Abdeljaber et al. [24] consisted of a methodology based on 1D CNN for damage localization in joints of a steel structure tested in the laboratory, as depicted in Fig. 4. This study holds great significance for introducing an important benchmark called the Qatar University Grandstand Simulator (QUGS), which has enabled numerous DL applications in the field of

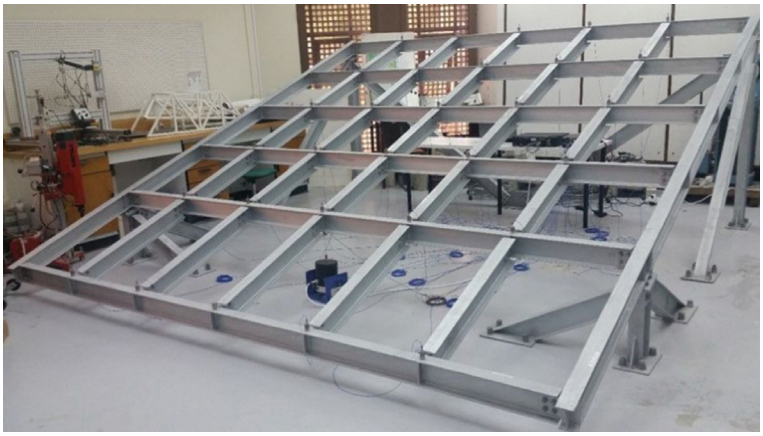


Fig. 4 Overview of steel structure QUGS [24]

SHM since then. Essentially, the damage entails loosening of the bolts connecting the joint between beams, simulating structural deterioration. The algorithm's objective is to correctly identify the position of any damaged joints. As part of the suggested methodology, a unique 1D CNN must be created and trained for every one of the 30 joints that have accelerometers mounted in them. It is the "responsibility" of each CNN_i to evaluate the condition of a particular joint i based solely on the raw acceleration data recorded at that joint. Since a single joint's network does not depend on support from other CNNs, the suggested damage detection technique is regarded as decentralized.

The training dataset was defined in such a way that the intact data from a specific joint contains the measured signals, while the other joints were intact and/or damaged as well, to minimize the influence of the damage effect from one joint on the others at the time of classification.

The methodology test is initiated by inducing damage at one or more locations (or keeping them intact) and applying a vibrator with random excitations. Then, the acceleration signals from each joint are measured and, afterwards, each signal is divided into samples of smaller intervals. Subsequently, the samples are normalized between -1 and 1 and used to feed their corresponding CNN for each joint. Thus, for each joint i, the probability of damage is given by $PoD_i = \frac{D_i}{T_i}$, where T_i is the total number of samples of interval signals processed by CNNi and D_i is the number of samples classified as damaged. Note that the final result is weighted by the intermediate results of each sample for greater representativeness of the data. Therefore, it is expected that the value of PoD_i is closer to 0 in intact joints and closer to 1 in damaged joints. Some of the results of this method are illustrated in Fig. 5.

The method was evaluated on 24 structural situations (1 intact case, 18 single damage cases, and 5 double damage instances) to assess the performance of the 30 CNNs that were generated. For the trained data, it was found that the 30 CNNs'

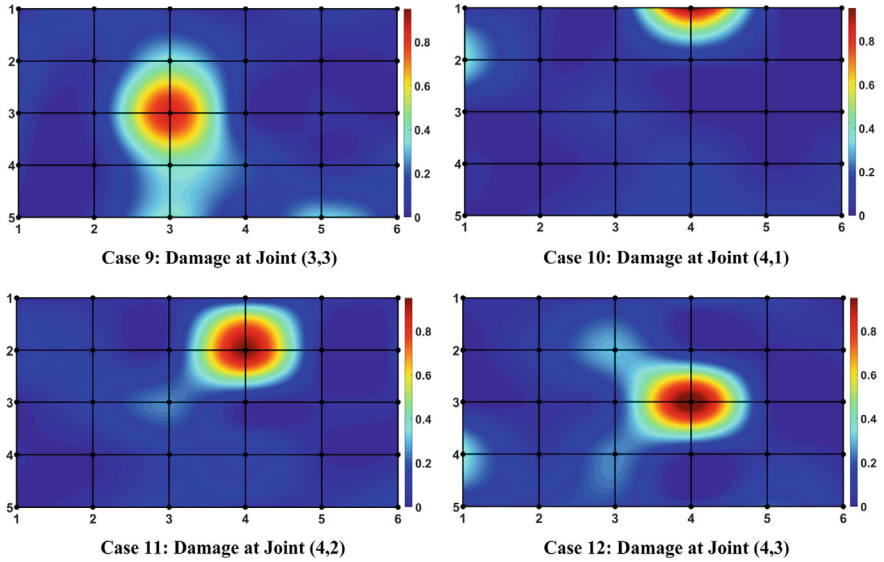


Fig. 5 Partial Results of Localization Using CNN [24]

average classification error was only 0.54%. The studies show that real-time SHM and structural damage detection work exceptionally well.

Long Short-Term Memory (LSTM) layers are used in a neural network architecture that Sony et al. [22] developed using the same QUGS dataset in addition to that of a real-scale bridge. These are looping networks that enable the persistence of data. This allows the DL approach to be trained and used to categorize time series signals into different damage levels and many classes with remarkable accuracy. In contrast, the application of the novel LSTM-based method in that study produced better damage localization outcomes than a comparable CNN.

Anomaly detection methods face challenges related to operating under variable conditions. Most existing methods are based on a stationary and constant assumption, which can lead to errors due to external effects or changes in the usage of structures. Therefore, Luo et al. [25] proposed a new method that allows for early fault detection under variable working conditions using Deep Learning with Stacked Auto-Encoders (AE) and Back Propagation Neural Network (BPNN) layers. A model was developed to automate the selection of impulse responses from large datasets of vibration collected during long-term operation. Based on the analysis of dynamic properties' similarity, an integrity index was constructed to indicate the slow and gradual deterioration process of the studied structure. The work validated the methodology in an experimental study involving a computer numerical control (CNC) machine, which does not have a constant operating pattern, allowing for the proper validation of the proposed methodology.

Similar to that, Finotti et al. [26] investigated a hybrid approach to characterize vibration signals with the goal of identifying anomalies in civil engineering structures. This approach used a Sparse Auto-Encoder (SAE) linked with SVM. A supervised classification model based on the conventional SVM was trained using the features that the SAE had directly extracted in an unsupervised manner from accelerations in the temporal domain. The suggested methodology was first evaluated using the structural reactions of a numerical beam model with simple support that was damaged to varying degrees. To confirm the SAE/SVM's efficacy on an actual structure, an analysis was conducted on experimental data obtained from the Várzea Nova viaduct located in João Pessoa, Brazil. In this study, dynamic tests were conducted on the viaduct before and after a reinforcement procedure. The method achieved over 99% accuracy in both cases.

A Deep Residual Network (DRN) was presented by Alazzawi and Wang [27] to automatically extract and learn features from raw acceleration signals. The effectiveness of this neural network was enhanced using a Bayesian optimization approach in terms of both parameter selection and accuracy. Damage was able to be identified, located, and measured using the recommended approach. Numerical and experimental applications were conducted on two 2-D frames and a bridge, achieving 100% accuracy on the employed test data.

Particularly, DL methods perform exceptionally well, but, as a limitation, they require a large amount of data to operate. To tackle this problem, Chamangard et al. [28] recently published a paper that suggested a method based on Transfer Learning (TL). There were two phases to the work. The first step aims to develop a compact CNN that can precisely detect structural faults given enough information from a numerical model. The problem of insufficient training data, or inadequate data for Structural Health Monitoring (SHM) of experimental and real-world structures, is examined in the second section. To address this problem, TL is employed. The final neural network obtains all the knowledge from a previously trained compact CNN on another structure, which is used as the source. Thus, the additional information obtained by this network can be used to compensate for the lack of data. In other words, when starting the application on a new structure, the use of TL tends to reduce the volume of required training data. The method was evaluated only for the classification of the existence or absence of damage (SHM level I). The compact CNN can achieve 100% accuracy when data are available for training. Furthermore, for the case of insufficient data, the use of a compact network as well as transfer learning provides considerable improvements (about 95%) in the accuracy of damage detection.

In the study by Luleci et al. [29], an alternative to the large volume of training data was developed based on the Wasserstein Deep Convolutional Generative Adversarial Networks (WDCGAN) with Gradient Penalty, also known as 1-D WDCGAN-GP. This allowed the researchers to generate additional synthetic acceleration data from the structure's damaged state, or damage data that had not been obtained through experimentation. The strategy fundamentally consists of two neural networks: one called the generator, which creates vibration signals, while the other, the discriminator, evaluates their authenticity. The generator begins to produce duplicates that

behave exactly like the originals at a given stage of learning. The created datasets were then confirmed by the authors utilizing both quantitative and qualitative techniques.

In a more recent advancement, Luleci et al. [30] employed an innovative variant of the GAN model, the 1-D CycleWDCGAN-GP. This model not only learns to generate data samples similar within a single domain but also controls the complex mapping between two data domains—undamaged and damaged. This model's capacity to extract "healthy" responses from a structure and convert them into damaged responses—thereby bridging the gap between the two domains—is a fascinating feature. Although the performance of the comparisons in the time domain has been quite good, the comparisons in the frequency domain could be improved even more, maybe with more varied training. To carry out this proposal, the researchers compiled a collection of dynamic response datasets from a steel structure, covering both damaged (loosening of joint bolts) and undamaged datasets, to train the CycleGAN variant model. Subsequently, they employed the undamaged dataset for translation into the damaged domain and vice versa. The performance indicators used in the evaluation process ensured that the generated results in both domains resembled the true domains, affirming the model's effectiveness in generating authentic data.

Despite their efforts, the outcome of the domain translation for the test joints was limited, as although it managed to translate the domain for the trained joint, it did not achieve the same success with the domains of the other joints. Thus, in an innovative study, Luceli et al. [31] investigated an enhanced version of the CycleGAN model to achieve higher accuracy. Some of the improvements involved: a more comprehensive training procedure to increase the model's domain knowledge, and the total generator loss function was supplemented with a new loss based on the frequency domain to better capture the frequency content of the data. With potential implications in predicting and simulating structural responses under a variety of situations, this groundbreaking discovery has opened fascinating new avenues for structural integrity monitoring. A key factor in improving safety protocols in critical infrastructures and furthering our understanding of structural behavior is the effective translation of reactions from healthy to unhealthy domains.

4 Model Interpretability Discussion

As observed, ML and DL techniques have been successfully applied in various real-world cases. However, the algorithms adopted in these studies for Vibration Structural Health Monitoring (VSHM) structures are often referred to as "black-box" models because they offer little information about the decision-making process [32]. One of the things preventing the widespread use of these techniques in industrial applications is the lack of interpretability and comprehension regarding how these models identify changes in structural integrity and how they use the features gathered from the data to create predictions, enabling interpretability in ML/DL-based VSHM methodologies can provide important evidence for reliable decision-making

in structural integrity monitoring applications. Specifically, the capacity to discriminate between abnormalities resulting from the impact of external operating variables and actual structural deterioration can assist lower false alarms and raise the system's overall accuracy.

According to Arrieta et al. [32], to define explainable AI models, two main groups of approaches have emerged: (a) transparent models, which offer direct explainability through the model itself or its structure, including decision trees, Bayesian models, k-nearest neighbors' models, and logistic or linear regression models. Some of these have already been mentioned in applications in this chapter. (b) Post-hoc explainability, divided into model-agnostic methods and model-specific methods, allows subsequent explainability of ML models. Model-agnostic methods provide explanations using general tools applicable to a wide variety of ML algorithms. On the other hand, model-specific methods focus on providing explanations for results obtained by specific ML algorithms, limited to some algorithms such as artificial neural networks and support vector machines.

Amin et al. [33] addressed the problem of understanding "black-box" CNN models in the context of condition monitoring for wind turbines operating in a variety of load scenarios, particularly those with turbulent winds. They applied strategies from Local Interpretable Model-Agnostic Explanations (LIME) to clarify the classifier's predictions. They were able to create an interpretable physics-based classifier as a result, which provided important insights into the logic underlying the model's judgments. The main goal of the study was to identify problems with the gearbox's rotating parts, which produce unique vibration signatures that accelerometers can measure. Their experiments were successful in identifying the flaws.

The goal of the study by Brusa et al. [34] was to determine the most pertinent aspects in the identification and classification of defects in condition monitoring systems for rotating machinery by using an interpretability technique called SHapley Additive exPlanations (SHAP). By gathering vibration data for various health states on an industrial bearing test rig housed in the Polytechnic University of Turin's Mechanical Engineering Laboratory, the researchers primarily looked at medium-sized bearings of industrial interest. SHAP was used to explain the diagnostic models SVM and k-Nearest Neighbour (kNN), which both obtained accuracy levels above 98.5% when utilizing it as a feature selection criterion.

The AI-based techniques discussed in this chapter have their characteristics summarized in Table 1. Additionally, it is understood as a general limitation of vibration-based damage detection methods the arrangement of sensors installed on the structure. Thus, if damage occurs in a location far from the sensors, it may not be detected. However, this disadvantage is becoming increasingly less significant with the development of sensor technologies, which are becoming cheaper and enabling their use in greater quantities. Additionally, several recent research efforts aim to optimize sensor placement [35] for increased efficiency in the detection process.

Table 1 Review of AI-based vibration damage detection methods

References	Approach	Tested structures	Type of damage	Strategies	SHM level
[14]	DS	Beam, 2-D frames and bridge	Holes, screw removal and cracks	Manually extracted features	I e II
[15]	DS	Beam, 2-D frames and bridges	Holes, cracks laying foundation	Manually extracted features	I, II e III
[17]	ML	Gantry, bars, and bridge	Nonlinearity, addition of mass and removal of interlocks	FS e SVM	I
[18]	ML	Gabbia Tower (Italy)	Earthquake	SVM	I
[19]	ML	2-D frame, truss and QUGC	Removing interlock and loosening screws	DT	I e II
[16]	ML	2-D frame	Nonlinearity	AR, SVM, GA, PSO	I e II
[20]	ML	Vehicle chassis and composite material bar	Additional mass and reduced stiffness	PSO	I e II
[34]	ML	Rotary machines	Bearing failures	SVM e kNN	I
[24]	DL	QUGS	Loosening screws	CNN	I e II
[25]	DL	CNC	Early failure	Stacked AE, BPNN	I
[26]	DL	Numerical beam and Várzea Nova viaduct in João Pessoa (Brazil)	Reduction of the modulus of elasticity and structural reinforcement	SAE e SVM	I
[22]	DL	Ponte e QUGS	Laying the foundation and loosening screws	LSTM	I, II e III
[27]	DL	Pórtico e ponte	Interlock Removal and Cuts	DRN	I, II e III
[28]	DL	QUGS e pontes	Cracks and loosening of screws	CNN e TL	I
[29–31]	DL	QUGS	Loosening of screws	CycleGAN	I
[33]	DL	Turbinas eólicas	Gear failures	CNN	I

5 Final Remarks

This research sought to present relevant and recent studies regarding the applications of AI in the context of SHM, using solely vibration data for structural damage detection. The present work focused on presenting the differences between ML and DL approaches, as well as their respective advantages and limitations. Furthermore, it

highlighted how researchers are currently dealing with these adverse factors, elucidating possibilities for future work. It is noted that in a ML approach, it is necessary to manually extract features from the acceleration signals, while a DL-based method performs the extraction automatically. It is understood that various methods implemented with AI have been successful in the paradigms of identifying the existence of damage, location, and/or quantification they aimed to solve.

However, the difficulty in collecting and labeling valid samples often leads to a small dataset in practical damage identification, reducing the performance of the trained model. For this issue, the implementation of finite element models, transfer learning, and even generative neural networks has been reported. Additionally, conventional deep learning models often lack a certain degree of physical interpretability. Such challenges have been mentioned and discussed as researchers have dealt with eliminating or mitigating these limitations, for example, using explainable AI techniques like LIME and SHAP.

A commonly observed aspect in the literature is the partitioning of acceleration signals to generate samples to be trained by the classifier with significant representativeness. Another similarity found is the recurrence of the tested structures. This is because they are benchmark-type structures, well-known and with data available to the scientific community. Below are listed the most used databases in articles in this area as noted by the authors:

- QUGS [24] (illustrated in Fig. 5);
- IASC-ASCE [36];
- Z24 viaduct [37];
- Three-Story 2-D frame—Los Alamos National Laboratory [38];
- Tianjin Yonghe Bridge [39].

The use of these datasets enables easy comparison with other detection methodologies, as well as providing greater credibility and reliability to the research, as they have already been validated previously. Therefore, the authors recommend, as a best practice, the use of one of the structures listed in future works investigating new SHM strategies. Nevertheless, new experimental vibration data from specific structures are always necessary and relatively difficult to find, such as offshore structures, aerospace, wind turbine generators, among others, revealing a wide range of industry sectors still underexplored.

Acknowledgements This work was financed by CAPES (Finance Code 0001), Conselho Nacional de Desenvolvimento Científico e Tecnológico—CNPq (Brazil)—Grants CNPq/FNDCT/MCTI 303982/2022-5 and 402533/2023-2; Fundação de Amparo à Pesquisa do Estado de Minas Gerais—FAPEMIG—Grant TEC PPM-00001-18.

References

1. Zhang, Y., Lei, Y.: Data anomaly detection of bridge structures using convolutional neural network based on structural vibration signals. Symmetry **13**(7), 1186 (2021)
2. Wang, T., Lu, G., Yan, P.: A novel statistical time-frequency analysis for rotating machine condition monitoring. IEEE Trans. Industr. Electron. **67**(1), 531–541 (2019)
3. Meixedo, A., Santos, J., Ribeiro, D., Calçada, R., Todd, M.: Damage detection in railway bridges using traffic-induced dynamic responses. Eng. Struct. **238**, 112189 (2021)
4. Yuan, F.G. (ed.): Structural Health Monitoring (SHM) in Aerospace Structures. Woodhead Publishing (2016)
5. Rytter, A.: Vibrational Based Inspection of Civil Engineering Structures. Department of Building Technology and Structural Engineering, Aalborg University (1993)
6. Moughty, J.J., Casas, J.R.: A state-of-the-art review of modal-based damage detection in bridges: development, challenges, and solutions. Appl. Sci. **7**(5), 510 (2017)
7. Ortiz Morales, F.A., Cury, A.A.: Analysis of thermal and damage effects over structural modal parameters. Struct. Eng. Mech.: Int. J. **65**(1), 43–51 (2018)
8. Cury, A., Crémona, C., Diday, E.: Application of symbolic data analysis for structural modification assessment. Eng. Struct. **32**(3), 762–775 (2010)
9. Dong, C.Z., Catbas, F.N.: A review of computer vision–based structural health monitoring at local and global levels. Struct. Health Monit. **20**(2), 692–743 (2021)
10. Tian, W., Cheng, X., Liu, Q., Yu, C., Gao, F., Chi, Y.: Meso-structure segmentation of concrete CT image based on mask and regional convolution neural network. Mater. Des. **208**, 109919 (2021)
11. Daneshjoo, Z., Shokrieh, M.M., Fakoor, M.: A micromechanical model for prediction of mixed mode I/II delamination of laminated composites considering fiber bridging effects. Theoret. Appl. Fract. Mech. **94**, 46–56 (2018)
12. Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., Inman, D.J.: A review of vibration-based damage detection in civil structures: from traditional methods to Machine Learning and Deep Learning applications. Mech. Syst. Signal Process. **147**, 107077 (2021)
13. Azimi, M., Eslamlou, A.D., Pekcan, G.: Data-driven structural health monitoring and damage detection through deep learning: state-of-the-art review. Sensors **20**(10), 2778 (2020)
14. Alves, V., Cury, A.: A fast and efficient feature extraction methodology for structural damage localization based on raw acceleration measurements. Struct. Control. Health Monit. **28**(7), e2748 (2021)
15. Alves, V., Cury, A.: An automated vibration-based structural damage localization strategy using filter-type feature selection. Mech. Syst. Signal Process. **190**, 110145 (2023)
16. Gui, G., Pan, H., Lin, Z., Li, Y., Yuan, Z.: Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. KSCE J. Civ. Eng. **21**, 523–534 (2017)
17. Ghiasi, R., Ghasemi, M.R., Chan, T.H.: Optimum feature selection for SHM of benchmark structures using efficient AI mechanism. Smart Struct. Syst. **27**, 623–640 (2021)
18. Finotti, R.P., de Souza Barbosa, F., Cury, A.A., Gentile, C.: A novel natural frequency-based technique to detect structural changes using computational intelligence. Procedia Eng. **199**, 3314–3319 (2017)
19. Mariniello, G., Pastore, T., Menna, C., Festa, P., Asprone, D.: Structural damage detection and localization using decision tree ensemble and vibration data. Comput.-Aided Civ. Infrastruct. Eng. **36**(9), 1129–1149 (2021)
20. Zacharakis, I., Giagopoulos, D.: Vibration-based damage detection using finite element modeling and the metaheuristic particle swarm optimization algorithm. Sensors **22**(14), 5079 (2022)
21. Cury, A., Ribeiro, D., Ubertini, F., Todd, M.D.: Structural health monitoring based on data science techniques. Cury, A. (Ed.). Springer (2022)

22. Sony, S., Gamage, S., Sadhu, A., Samarabandu, J.: Vibration-based multiclass damage detection and localization using long short-term memory networks. In: Structures, vol. 35, pp. 436–451. Elsevier (Jan 2022)
23. Cireşan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep, big, simple neural nets for handwritten digit recognition. Neural Comput. **22**(12), 3207–3220 (2010)
24. Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., Inman, D.J.: Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. J. Sound Vib. **388**, 154–170 (2017)
25. Luo, B., Wang, H., Liu, H., Li, B., Peng, F.: Early fault detection of machine tools based on deep learning and dynamic identification. IEEE Trans. Industr. Electron. **66**(1), 509–518 (2018)
26. Finotti, R.P., Barbosa, F.D.S., Cury, A.A., Pimentel, R.L.: Numerical and experimental evaluation of structural changes using sparse auto-encoders and SVM applied to dynamic responses. Appl. Sci. **11**(24), 11965 (2021)
27. Alazzawi, O., Wang, D.: A novel structural damage identification method based on the acceleration responses under ambient vibration and an optimized deep residual algorithm. Struct. Health Monit. **21**(6), 2587–2617 (2022)
28. Chamangard, M., Ghodrati Amiri, G., Darvishan, E., Rastin, Z.: Transfer learning for CNN-based damage detection in civil structures with insufficient data. Shock. Vib. **2022** (2022)
29. Luleci, F., Catbas, F.N., Avci, O.: Generative adversarial networks for labeled acceleration data augmentation for structural damage detection. J. Civ. Struct. Heal. Monit. **13**(1), 181–198 (2023a)
30. Luleci, F., Catbas, F.N., Avci, O.: CycleGAN for undamaged-to-damaged domain translation for structural health monitoring and damage detection. Mech. Syst. Signal Process. **197**, 110370 (2023b)
31. Luleci, F., Avci, O., Catbas, F.N.: Improved undamaged-to-damaged acceleration response translation for Structural Health Monitoring. Eng. Appl. Artif. Intell. **122**, 106146 (2023c)
32. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities, and challenges toward responsible AI. Inf. Fusion **58**, 82–115 (2020)
33. Amin, A., Bibo, A., Panyam, M., Tallapragada, P.: Wind Turbine gearbox fault diagnosis using cyclostationary analysis and interpretable CNN. J. Vib. Eng. Technol., 1–11 (2023)
34. Brusa, E., Cibrario, L., Delprete, C., Di Maggio, L.G.: Explainable AI for machine fault diagnosis: understanding features' contribution in machine learning models for industrial condition monitoring. Appl. Sci. **13**(4), 2038 (2023)
35. Tan, Y., Zhang, L.: Computational methodologies for optimal sensor placement in structural health monitoring: a review. Struct. Health Monit. **19**(4), 1287–1308 (2020)
36. Johnson, E.A., Lam, H.F., Katafygiotis, L.S., Beck, J.L.: Phase I IASC-ASCE structural health monitoring benchmark problem using simulated data. J. Eng. Mech. **130**(1), 3–15 (2004)
37. Maeck, J., De Roeck, G.: Damage assessment using vibration analysis on the Z24-bridge. Mech. Syst. Signal Process. **17**(1), 133–142 (2003)
38. Figueiredo, E., Park, G., Figueiras, J., Farrar, C., Worden, K.: Structural Health Monitoring Algorithm Comparisons using Standard Data Sets (No. LA-14393). Los Alamos National Lab. (LANL), Los Alamos, NM (United States) (2009)
39. Li, H., Li, S., Ou, J., Li, H.: Reliability assessment of cable-stayed bridges based on structural health monitoring techniques. Struct. Infrastruct. Eng. **8**(9), 829–845 (2012)

Optimizing Tuned Mass Damper by Examining Displacement Ratios with and Without TMD System

Muhammed Çoşut, Sinan Melih Nigdeli, and Gebrail Bekdaş

Abstract That control systems have been used worldwide is explained herein; moreover, Tuned Mass Damper optimization is conducted using displacement with and without TMD according to FEMA far-fault data (FEMA P-695, Quantification of Building Seismic Performance Factors. Washington) which consists of destructive earthquake records throughout the time. This record includes 44 different earthquakes observed in various areas in the world. Choosing objective function as the ratio of TMD's displacement over without TMD's displacement allows us to reach more appropriate results of TMD system due to the usage of every earthquake records' ratio. What is more, in order to delineate as well as elucidate the results of this study, different studies' displacement results and the ratio of differences between control system usage and no control system usage are taken into consideration.

Keywords Tuned Mass Damper · Displacements of earthquakes · Ratio optimization · Comparative study

1 Introduction

It is observed that there is a myriad of differences in building systems, elements and structural materials from past to present. These differences have been and continue to be realised as a result of observations, experiments and studies, such as the correction of some structural defects encountered and the completion of the life of the building systems in a safer way under loads. In this scope, in order to limit the displacements of high-rise buildings and prevent the deterioration of the irregularities of the structures to be applied in areas with high earthquake risk, building control systems developed over time with different properties are preferred [2]. One of the most important

M. Çoşut · S. M. Nigdeli · G. Bekdaş (✉)
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_8

features that distinguish building control systems from each other is the difference in the materials used in the system. While some systems contain a computer system with a control mechanism [3] within itself, some are created by using different materials in the system, and some building control systems will prevent damage to the structure as a result of the interactions between the materials. Such control systems should be added to the systems with appropriate designs and the analyses of the structure and the system for which the structure control systems are desired to be designed should be carried out appropriately and accurately and the suitability should be analysed. Due to the expense of these systems, attention is paid to creating designs for maximum performance in the most accurate location in the structure and with minimum features. Machine learning [4] and metaheuristic algorithms are frequently adopted to implement appropriate applications in a variety of different areas of building control systems.

Several studies have been completed and sought throughout the times so as to enhance the capacity of structures and accommodate state of art constructions. Review research pertaining to different types of structural control systems [5, 6]. TMD optimization in terms of displacement, period, frequency, damping along earthquake features was conducted separately, and these were all done by using different metaheuristic algorithms and hybrid algorithms [7–14].

In this study, period and damping are set up as variables for the equations to reach Tuned Mass Damper properties which is conducted in MATLAB & Simulink. Jaya Algorithm which is one of the metaheuristic algorithms is chosen, and the objective function is generated for every earthquake ratio of TMD's displacement over without TMD's displacement by using FEMA earthquake records. Regarding objective function, the fundamental aim is to decrease the displacement ratio in every earthquake; hence, systems will be more durable and resistant.

2 Vibration

Vibration in the structures might originate from different sources such as wind loading, earthquakes as well as machines. Repetitive motions cause vibration in the systems, and sometimes cannot resist overloading; therefore, damping systems can be used in the structure so as to control structural motions [15]. Free vibration, forced vibration, damped vibration, undamped vibration, harmonic vibration and torsional vibration are given as examples of the types of vibration which all have different effects and properties on the structures. These vibrations are generally based on structural features, force features, damping features of the structures and soil features. To illustrate, undamped vibration sustains the effect of vibration continuously in the structures whereas damped vibration amplitude always decreases over time and finally reaches no displacement. In some cases, maximum vibration in the constructions with maximum amplitude is reduced by employing control mechanisms.

Furthermore, period and frequency hinge on equations of vibration, so it should be known that period is the required time for one periodic motion completed, as well as frequency is the number of cycles per unit of time. The period is affected more by mass along with the stiffness of the systems. Equations 1 and 2 are used to find the period and frequency, respectively. w_n denotes the natural frequency of vibration

$$T = \frac{2 \times \pi}{w_n} \tag{1}$$

$$w = \frac{1}{T} \tag{2}$$

$$k = \frac{12 \times E \times I}{L^3} \tag{3}$$

$$w_n^2 = \frac{k}{m} \tag{4}$$

Structure stiffness and natural frequency of structure can be calculated by Eqs. 3 and 4, respectively. E is Young's Module, I is the inertia of the moment, L is the length of the members as well as m is the mass of the structure.

3 Structural Control System

Building control systems with different characteristics have been developed over time and these systems are frequently used to prevent structural defects that may occur in the irregularity and rigidity of the structure. Although the structures are built in accordance with the regulations of the country in which they are located, sometimes design errors can be made during the application phase, which can lead to damage to the structure and even loss of life. In order to overcome such hazards, building control systems that can be integrated into the building afterwards or can be incorporated into the system during the initial design phase of the building are considered preferable. These control systems have different groups that are active control systems, passive control systems, hybrid control systems and semi-active control systems. Active control systems include active tuned mass damper, active tendon control and active rigidity control. Systems frequently used as passive control systems: friction type damper [16], yielding metal damper [17], viscoelastic damper [18], viscous damper [16], passive tuned mass damper as well as tuned liquid damper [19]. In addition, hybrid and semi-active control systems combined with other control systems increase the capacity of structure under loading. Applications of structural control systems are shown in Table 1.

Active control systems which were exhibited by Zuk [20, 21] in 1968 aim to ensure that the structure always remains on the safe side by ensuring that the control system responds according to the incoming loads under earthquake and wind loads, which are seismic loads that will affect the structure, thanks to some of the mechanical

Table 1 Applications of structural control systems

Structures	Location	Types of systems
Millennium Tower	Japan	TMD Seismic Isolator
Shangai Tower	China	TMD
CN Tower		TMD
Millenium Bridge	UK	TMD
Hancock Tower	USA	TMD
The Shard	UK	Seismic Isolator Dampers
The Bow	Canada	TMD
U.S. Bank Tower	USA	TMD
One Rincon Hill	USA	LMD
Air Traffic Control Tower	India	TMD
Tiwest Rutile Plant Chimney	Australia	TMD

devices it contains. The most important and superior feature compared to passive control systems is that it can be resisted by changing the load value in the system, as well as by changing other features to protect structures. Another important feature is that it does not need a lot of space in the structure like passive control systems. However, one should keep in underlying in mind that sometimes these control devices in the system may delay in responding to the load and this may cause some damage to the structure, albeit partially. In addition, the implementation of these systems will be a little more expensive due to the valuable materials required for the installation.

Passive control systems (see Figs. 2 and 3) are frequently preferred in building systems since they can be attached to the construction in an effortless manner. Applications are started by selecting the type according to the location where these elements will be placed in the structure. For TMD (see Fig. 1), it was first created by Frahm [22] in 1909, and after studies, Ormondroyd and Den Hartog [23] ameliorated it. The results of Der Hartog's studies [24, 25] were about the details of when TMD attaches the structures.

Passive control systems are encountered with different applications and designs in order to keep the structure on the safe side. In order to increase the resistance to loads acting on the structure, spring and damping mechanism, distribution of energy by systems, application of isolation elements, use of properties of materials such as friction, cross-sections of design elements and moment of inertia are utilised.

Figure 4 demonstrates the Multi Degree of Freedom (MDOF) with a Tuned Mass Damper (TMD). The essence of these applications is that it should be noted that displacement of initial, floors and also damping influence the system solutions. Regarding the damping session, the displacement of TMD impacted by initial displacement, and the latest floor displacement with TMD displacement should be taken into consideration in the algorithms so as to reach a more applicable and appropriate design for the structures.

(a)

(b)

(c)

Fig. 1 TMD applications: **a** CN Tower, **b** Hancock Tower, **c** Millennium Bridge. http://upload.wikimedia.org/wikipedia/commons/d/dd/Toronto_-_ON_-_CN_Tower_-_Antennenspitze.jpg, http://upload.wikimedia.org/wikipedia/commons/5/56/John_Hancock_Tower.jpg

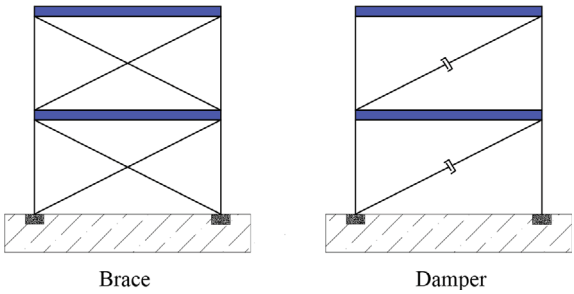


Fig. 2 Different types of control

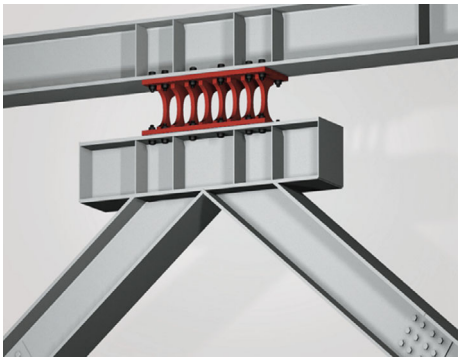


Fig. 3 Metallic yield damper (http://en.roadjz.com/show.asp?id=19#:~:text=When%20the%20structural%20suffered%20by,by%20the%20seismic%20events%20sufficiently)

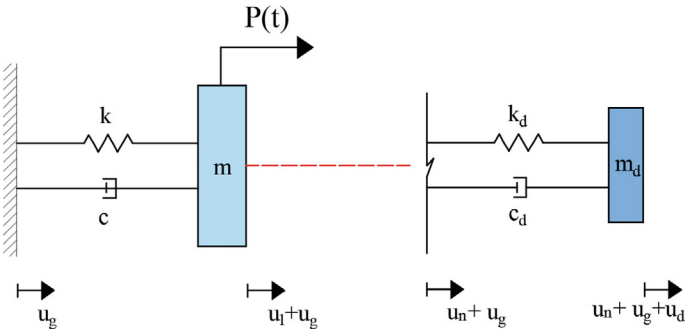


Fig. 4 TMD application in the MDOF system

In the case TMD (see Fig. 1) is desired to be utilised, it is generally placed on the top floor of the structure and aims to keep the displacements of the structure at a certain level under seismic loads. The reason for placing it on the top floor of the structure is; structures without a control system will have the highest displacement value at the top floor under loads, provided that there are no irregularities in the construction. It is therefore requested to limit the displacements in that section by placing the systems on the top floor. Obviously, some structures have structural irregularities such as soft

floors, weak floors, etc., and this will cause relative displacements to occur more on different floors rather than on the top floor. Hence, before designing control systems for a structure, it should first be checked whether these systems are proper for that configuration.

4 Optimization

That optimization of a myriad of examples and studies in many areas which are economy, time management, civil engineering, transportation along with mechanics has been conducted recently explicitly observed, and designers therefore aim to create more suitable products or systems in order to maintain cost management, sustainability as well as customers' desires [26]. In this regard, minimum cost, minimum CO_2 emission, minimum displacement of structure, maximum speed of machines, optimum mass for Tuned Mass Damper as well as optimum damping coefficient with period for dampers can be computed as objective functions in terms of the problems, and sometimes these objective functions can be combined so as to underpin both requirements at the same time. Artificial intelligence, machine learning, deep learning [27] and metaheuristic algorithms are ubiquitously employed in numerous studies to attain more consistent and accurate results. Regarding metaheuristic algorithms, the Jaya algorithm which was generated by Rao [28] is employed herein.

5 Numerical Example

Tuned Mass Damper is applied in the structure of which features are given Table 2; moreover, TMD range of variables and constraint are given in the same table.

In order to solve TMD system, mass, stiffness and damping should be found according to the affected loads (Eqs. 5–9). These equations are implemented for the

Table 2 Structure and TMD properties

	Symbol	Unit	Value and range
Structure	m: mass	kg	2924
	c: damping coefficient	Ns/m	1581
	k: stiffness coefficient	N/m	1,390,000
TMD	m_d: mass	kg	0.05 × (Structure Mass)
	Damping ratio		0.01–05
	T: period	s	0.5 × (Structure Period) 1.5 × (Structure Period)
	Stroke capacity		$g(X) = \frac{\max(\lvert xd - xN \rvert)TMDwith}{\lvert xN \rvert TMDwithout} \leq st\_\max$

single degree of freedom system, and if the degree of freedom system is increased, meanwhile, equations should be changed in terms of how many degrees of freedom system has.

$$[M] = \begin{bmatrix} m & 0 \\ 0 & m_d \end{bmatrix} \tag{5}$$

$$[K] = \begin{bmatrix} k + k_d & -k_d \\ -k_d & k_d \end{bmatrix} \tag{6}$$

$$[C] = \begin{bmatrix} c + c_d & -c_d \\ -c_d & c_d \end{bmatrix} \tag{7}$$

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = 0 \tag{8}$$

Provided that the system is affected by seismic loading, this loading should be considered in the equations (Eq. 9) to reach the precise result.

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F\} \tag{9}$$

Table 3 displays the FEMA far-fault data with its component, years. What is more, system displacement is computed with TMD and without TMD in terms of every earthquake record. Duzce/Turkey with 0.0493 m displacement which was recorded as the fifth is observed the most displacement according to Jaya parameters as the iteration number was taken as 100, and the population number was taken as 10 to interpret former studies [10, 29]. In general, it can obviously be noticed that maximum displacement without TMD was found different as more displacement was recorded in this study. Nevertheless, there is no need to focus on these displacement differences. The primary importance is to compare how ratio displacement is reduced by employing TMD.

So as to consolidate the aforementioned idea, Table 4 which is related to former study results and this investigation's outcomes is generated for comparison between objective functions, mean change displacement ratio of with and without TMD, maximum change displacement ratio of with and without TMD, standard deviation, period of TMD and damping ratio of TMD. The mean of displacement differences for this study and Cosut et al. study [10] are calculated as 37.42% and 23.87%, respectively, which means when objective function clarified as the ratio between differences between with TMD and without TMD displacement gives more suitable results compared to objective function reducing maximum displacement. In addition, the maximum ratio of these is found 54.96% along with 42.33 in the name order. When it comes to standard deviation, they got relatively close results. The period and damping ratio of TMD in the findings of this study is 0.4199 s and 0.2478, respectively, while the period and damping ratio of TMD in Cosut et al. investigation [10] are 0.3423 s and 0.4758, respectively.

Table 3 FEMA far-fault data and displacement results

Earthquake name	Year	Components	Maximum displacement			
			With TMD	Without TMD	With TMD [10]	Without TMD [10]
Northridge	1994	1	0.0204	0.0352	0.0169	0.0271
		2	0.0303	0.0570	0.0263	0.0377
Northridge	1994	1	0.0185	0.0375	0.0171	0.0229
		2	0.0241	0.0455	0.0189	0.0236
Duzce, Turkey	1999	1	0.0493	0.0757	0.0435	0.0455
		2	0.0213	0.0473	0.0211	0.0213
Hector Mine	1999	1	0.0114	0.0162	0.0113	0.0160
		2	0.0190	0.0332	0.0183	0.0304
Imperial Valley	1979	1	0.0128	0.0197	0.0129	0.0206
		2	0.0185	0.0258	0.0175	0.0201
Imperial Valley	1979	1	0.0359	0.0456	0.0356	0.0617
		2	0.0286	0.0393	0.0275	0.0288
Kobe, Japan	1995	1	0.0280	0.0476	0.0281	0.0350
		2	0.0270	0.0435	0.0267	0.0463
Kobe, Japan	1995	1	0.0096	0.0199	0.0074	0.0079
		2	0.0114	0.0203	0.0095	0.0115
Kocaeli, Turkey	1999	1	0.0153	0.0292	0.0118	0.0146
		2	0.0312	0.0507	0.0311	0.0361
Kocaeli, Turkey	1999	1	0.0059	0.0082	0.0060	0.0087
		2	0.0078	0.0097	0.0078	0.0097
Landers	1992	1	0.0095	0.0183	0.0096	0.0148
		2	0.0132	0.0167	0.0120	0.0198
Landers	1992	1	0.0263	0.0366	0.0244	0.0287
		2	0.0229	0.0393	0.0198	0.0212
Loma Prieta	1989	1	0.0456	0.0630	0.0424	0.0732
		2	0.0171	0.0330	0.0163	0.0212
Loma Prieta	1989	1	0.0270	0.0419	0.0271	0.0400
		2	0.0278	0.0383	0.0237	0.0289
Manjil, Iran	1990	1	0.0262	0.0370	0.0260	0.0277
		2	0.0345	0.0432	0.0339	0.0396
Superstition Hills	1987	1	0.0138	0.0238	0.0137	0.0204
		2	0.0093	0.0159	0.0095	0.0151
Superstition Hills	1987	1	0.0195	0.0347	0.0179	0.0250
		2	0.0146	0.0248	0.0144	0.0169

(continued)

Table 3 (continued)

Earthquake name	Year	Components	Maximum displacement			
			With TMD	Without TMD	With TMD [10]	Without TMD [10]
Cape Mendocino	1992	1	0.0224	0.0395	0.0222	0.0302
		2	0.0378	0.0625	0.0376	0.0469
Chi-Chi, Taiwan	1999	1	0.0190	0.0262	0.0189	0.0275
		2	0.0212	0.0418	0.0142	0.0191
Chi-Chi, Taiwan	1999	1	0.0265	0.0480	0.0263	0.0275
		2	0.0206	0.0355	0.0196	0.0323
San Fernando	1971	1	0.0134	0.0190	0.0133	0.0195
		2	0.0096	0.0135	0.0088	0.0110
Friuli, Italy	1976	1	0.0205	0.0301	0.0205	0.0300
		2	0.0162	0.0297	0.0163	0.0205

Table 4 Studies result

Explanation	Reducing ratio of TMD's displacement over without TMD's displacement for every earthquake record	Reducing max displacement [10]
Mean change between displacements with and without TMD (%)	37.42	23.87
Maximum change between displacements with and without TMD (%)	54.96	42.33
Standard deviation	0.095	0.12
Period	0.4199	0.3423
Damping ratio	0.2478	0.4758

Table 5 illustrates the studies of TMD maximum displacement. All these 3 investigations have the same properties as metaheuristic algorithms, variables, constraints along with constants. With respect to this study which was run in MATLAB & Simulink one time and maximum displacement was recorded as 0.0493 m. Nevertheless, the other studies were run 10 times and saved all maximum displacement. Hence, for reducing max displacements studies are demonstrated as intervals which are 0.0418–0.0659 m [10] and 0.0466–0.0707 m [29]

Table 5 Displacement of studies

Explanation	Maximum displacement
Reducing ratio of TMD's displacement over without TMD's displacement for every earthquake record	0.0493
Reducing max displacement [10]	0.0418–0.0659
Reducing max displacement [29]	0.0466–0.0707

6 Conclusion

That using a structure control system in the structures reduces the maximum displacement is an undeniable fact. Nevertheless, it should be noted that structure control systems should be chosen according to the structures' properties. Thereafter, designs have to support by the analyses. TMD is one of the structural control systems utilised herein to reduce displacement of the structure, and period along with damping of TMD were specified as the variables. The stroke capacity of TMD is taken into consideration as a constraint. FEMA far-fault data which includes 44 destructive earthquakes was implemented for every earthquake to find all displacement with and without TMD. The objective function was generated as the ratio of TMD's displacement over without TMD's displacement for every earthquake in the optimization process, and they were saved to compare relations between earthquakes and current research. Comparison of different studies' results empowers the up-to-date studies more rigorous and accurate as an assessment. When looking at the other studies, it can clearly be seen that although the finding displacement without TMD is less than this study's results, this study with different objective function which is the displacement ratio between TMD system displacements and no TMD system displacement enables structure to reduce its displacement more than current investigation with different variables values. As for maximum displacement comparison, some studies have 10 runs, which is why, they were taken as an interval to show the range of displacement. This main difference between these intervals can be affected by TMD damping ratio as well as the period. This study's maximum value was found between other intervals. Ongoing and future studies for TMD require to full focus on different objective functions to establish the most appropriate design.

References

1. FEMA P-695: Quantification of Building Seismic Performance Factors. Washington
2. Spencer, B.F., Jr., Nagarajaiah, S.: State of the art of structural control. J. Struct. Eng. **129**(7), 845–856 (2003)
3. Nishimura, I., Kobori, T., Sakamoto, M., Koshika, N., Sasaki, K., Ohrui, S.: Active tuned mass damper. Smart Mater. Struct. **1**(4), 306 (1992)
4. Yucel, M., Bekdaş, G., Nigdeli, S.M., Sevgen, S.: Estimation of optimum tuned mass damper parameters via machine learning. J. Build. Eng. **26**, 100847 (2019)

5. Housner, G.W., Bergman, L.A., Caughey, T.K., Chassiakos, A.G., Claus, R.O., Masri, S.F., Skelton, R.E., Soong, T.T., Spencer, B.F., Jr., Yao, J.T.P.: Structural control: past, present and future. J. Eng. Mech., ASCE **123**(9), 897–971 (1997)
6. Symans, M.D.: Semiactive control systems for seismic protection of structures: a state-of-the-art review. Eng. Struct. **21**, 469–487 (1999)
7. Leung, A.Y.T., Zhang, H.: Particle swarm optimization of tuned mass dampers. Eng. Struct. **31**(3), 715–728 (2009)
8. Zuo, L., Nayfeh, S.A.: Optimization of the individual stiffness and damping parameters in multiple-tuned-mass-damper systems. J. Vib. Acoust. **127**(1), 77–83 (2005)
9. Lu, Z., Chen, X., Zhou, Y.: An equivalent method for optimization of particle tuned mass damper based on experimental parametric study. J. Sound Vib. **419**, 571–584 (2018)
10. Çoşut, M., Nigdeli, S.M., Bekdaş, G.: Structural control systems and tuned mass damper optimization by using Jaya and hybrid algorithms. In: Hybrid Metaheuristics in Structural Engineering: Including Machine Learning Applications, pp. 87–110. Springer Nature Switzerland, Cham (2023)
11. Kamgar, R., Samea, P., Khatibinia, M.: Optimizing parameters of tuned mass damper subjected to critical earthquake. Struct. Design Tall Spec. Build. **27**(7), 1460 (2018)
12. Zuo, L., Nayfeh, S.A.: Minimax optimization of multi-degree-of-freedom tuned-mass dampers. J. Sound Vib. **272**(3–5), 893–908 (2004)
13. Bekdaş, G., Nigdeli, S.M.: Estimating optimum parameters of tuned mass dampers using harmony search. Eng. Struct. **33**(9), 2716–2723 (2011)
14. Bekdaş, G., Nigdeli, S.M.: Mass ratio factor for optimum tuned mass damper strategies. Int. J. Mech. Sci. **71**, 68–84 (2013)
15. Bachmann, H.: Vibration Problems in Structures: Practical Guidelines. Springer Science & Business Media (1995)
16. De Domenico, D., Ricciardi, G.: Earthquake protection of structures with nonlinear viscous dampers optimized through an energy-based stochastic approach. Eng. Struct. **179**, 523–539 (2019)
17. Moreschi, L.M., Singh, M.P.: Design of yielding metallic and friction dampers for optimal seismic performance. Earthquake Eng. Struct. Dynam. **32**(8), 1291–1311 (2003)
18. Xu, Z.D., Shen, Y.P., Zhao, H.T.: A synthetic optimization analysis method on structures with viscoelastic dampers. Soil Dyn. Earthq. Eng. **23**(8), 683–689 (2003)
19. Ocak, A., Nigdeli, S.M., Bekdaş, G., Kim, S., Geem, Z.W.: Adaptive harmony search for tuned liquid damper optimization under seismic excitation. Appl. Sci. **12**(5), 2645 (2022)
20. Datta, T.K.: A state-of-the-art review on active control of structures. ISET J. Earthq. Technol. **40**(1), 1–17 (2003)
21. Zuk, W.: Kinetic structures. Civ. Eng. **39**(12), 62–64 (1968)
22. Frahm, H.: Device for Damping Vibration of Bodies, US Patent, 989958 (1909)
23. Ormondroyd, J., Den Hartog, J.P.: The theory of the dynamic vibration absorber. Trans. Am. Soc. Mech. Eng. **50**, 9–22 (1928)
24. Den Hartog, J.P.: Mechanical Vibrations, 4th ed., McGraw-Hill, New York (NY), USA (1956)
25. Elias, S., Matsagar, V.: Research developments in vibration control of structures using passive tuned mass dampers. Annu. Rev. Control. **44**, 129–156 (2017)
26. Bekdaş, G., Nigdeli, M.N., Yücel, M., Kayabekir, A.E.: Yapay Zeka optimizasyon Algoritmaları ve Mühendislik Uygulamaları, Seçkin Yayıncılık, Ankara (2021). ISBN: 978-975-02-6637-9
27. Aydin, Y., Bekdaş, G., Işıkdağ, Ü., Nigdeli, S.M.: The state of art in machine learning applications in civil engineering. In: Hybrid Metaheuristics in Structural Engineering: Including Machine Learning Applications, pp. 147–177 (2023)
28. Rao, R.: Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. **7**(1), 19–34 (2016)
29. Çoşut, M., Nigdeli, S.M., Bekdaş, G.: The sensibility of Jaya algorithm on tuned mass damper optimization. In: 6th International Conference on Intelligent Computing & Optimization (ICO2023), 27–28 April, Hua Hin Thailand

Evaluation of Predictive Models for Mechanical Properties of Earth-Based Composites for Sustainable Building Applications

Ifeyinwa Ijeoma Obianyo, Azikiwe Peter Onwualu, and Assia Aboubakar Mahamat

Abstract Increase in human population have placed a high demand on civil engineering projects leading to high cost of conventional building composites. Adequate utilization of earth-based composites such as bricks and stabilized soils for building and road construction have the potential of bridging the infrastructure deficit gap across the globe. To save resources and promote the effective use of earth-based composites as alternative affordable and sustainable building composites, this study aims to evaluate models for predicting the mechanical properties of earth-based composites for sustainable building applications. Desk study was employed to evaluate existing models for predicting the mechanical properties of earth-based composites. Several predictive models used for the prediction of the mechanical properties of earth-based composites were identified and analyzed. The performances of the identified models were classified using the respective values of their R-squared, the root mean square errors, coefficient of correlation, root absolute error, mean absolute error and root relative squared error. The model that outperformed other identified models was selected and recommended as the best-fit model for the prediction of the mechanical properties of earth-based composites. This study provides valuable insights to civil engineers and other professionals in the construction industry on efficient utilization of earth-based composites for sustainable building applications.

Keywords Predictive models · Mechanical properties · Environmental sustainability · Earth-based composites · Building projects

I. I. Obianyo (✉) · A. A. Mahamat
Department of Civil Engineering, Nile University of Nigeria, Abuja, Nigeria
e-mail: ifeyinwa.obianyo@nileuniversity.edu.ng

A. P. Onwualu
Department of Mechanical Engineering, African University of Science and Technology Abuja, Abuja, Nigeria

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_9

1 Introduction

Earth-based composites have been used in construction of various projects due to its affordability and sustainability. Over the years, conventional building materials such as concrete has been used in building projects [1]. Civil Engineers are faced with the challenge of striking a balance between functionality, aesthics and environmental sustainability [2]. There is therefore, the need to seek alternative building materials that are functional and eco-friendly. As a sustainable alternative building materials, earth-based composites have been used in the developed and developing countries for the development of low-cost housing and roads projects [3]. Utilization of earth-based composites has the potentials of addressing the high demands of infrastructural projects and maintaining environmental sustainability.

Earth-based composites are building materials that produced with earth-based materials which include soil, rock and minerals. These earth-based materials are naturally occurring materials. Developing buiding composites with earth-based materials ensures the sustainability of the earth through the reduction in the consumption of concrete materials with high carbon footprint. Although some factors such aesthetics and strength have hindered the utilization of earth-based composites in mega building projects [4], researchers have explored ways of improving the strength of earth-based composites through stabilization [5, 6]and the use of addictives [7] In recent times, earth-based composites with improved strength and aesthetic have find application in various building projects.

Predictive modelling of mechanical properties of earth-based composites is essential in promoting the utilization of earth-based composites for diverse building projects. The use of predictive models for stength prediction do not only save time but also save energy and resources [8, 9] Several predictive and machine learning models have been used to model the mechanical properties (compressive strength, flexural strength, etc.) of earth-based composties and they include Support Vector Machines [10] Random Forest [11]; Artificial Neural Network [12]; Radial Basis Function (RBF) neural networks [13]; and Response Surface Methodology [14].

This chapter is organized into four main sections which include Sect. 1 (introduction), Sect. 2 (predictive machine learning models for mechanical properties of earth-based composites), Sect. 3 (performance of predictive models for mechanical properties of earth-based composites) and Sect. 4 (conclusion). Section 1 gives the background on the topic of this chapter while Sect. 2 presents the predictive machine learning models for mechanical properties of earth-based composites. The evaluation of performance of predictive models for mechanical properties of earth-based composites was presented in Sect. 3 while Sect. 4 as the final section provides a brief conclusion of the chapter.

2 Predictive Machine Learning Models for Mechanical Properties of Earth-Based Composites

Machine learning models have find several applications in the field of Civil Engineering[15] These applications include structural health monitoring of buildings [16–18]; predicting engineering materials behavior [19, 20]; construction project lifecycles [21]; structural design and performance assessment [22] and classifying and solving complex mathematical problems in civil engineering [23, 24]. The prediction of the mechanical performane of earth-based composites is crucial to the optimal design and application of these materials in building projects. Several models that have been used for the prediction of the mechanical performane of earth-based composites are discussed below.

2.1 Predictive Machine Learning Models for Mechanical Properties of Earth-Based Composites

Mechanical properties of earth-based composites include compressive strength, flexural strength, tensile strength, California bearing ratio, shear strength, hardness and maximum dry density. These mechanical properties give an indication of the performance of the earth-based composites in building applications. Table 1 shows the predictive machine learning models that have been used for the prediction of the mechanical properties of earth-based composites.

Table 1 Predictive machine learning models used for the prediction of the mechanical properties of earth-based composites

References	Earth-based composites	Mechanical properties	Name of models used
[13]	Stabilised soil	Maximum dry density and Optimum moisture content	Radial Basis Function (RBF) neural networks
[25]	Bamboo fiber-reinforced, palm oil-based resin composites	Tensile strength	Gradient boost decision tree
[26]	Natural pozzolana and limestone blended concrete	Flexural Strength	Artificial neural networks
[27]	Metakaolin concrete	Splitting tensile strength	Gradient boosting decision tree

Table 2 Evaluation of model input and output variables for predicting the mechanical properties of earth-based composites

Input variables	Output variable	Model name	R^2	References
W/C ratio, cement content, water content, sand content, sulfate concentration, wetting temperature, drying time, drying temperature, wetting time and exposure time	Compressive strength	Artificial neural networks and support vector machine	99	[28]
Plastic, sand, sand, size, fiber percentage, fibre length, fibre diameter, and tensile strength of the fibre	Compressive strength	Gene expression programming (GEP)	88	[29]

3 Performance of Predictive Models for Mechanical Properties of Earth-Based Composites

3.1 Evaluation of Model Input and Output Variables for Predicting the Mechanical Properties of Earth-Based Composites

In the modelling of the mechanical properties of earth-based composites, establishment of the dependent (output variables) and independent variables (input variables) is critical to the overall performance of the model used. Some of the input variables used to predict the mechanical properties of earth-based composites include Cement (C), fine aggregates (FA), coarse aggregates (CA), water-cement ratio (W/C), marble powder (MP), and curing days (CD). Table 2 presents the evaluation of the model input and output variables for predicting the mechanical properties of earth-based composites.

3.2 Comparison of the Performance of Models Used for Earth-Based Composites

The performance of predictive models can be ascertained using the root mean square error (RMSE), coefficient of correlation (CC), relative absolute error (RAE), mean absolute error (MAE) and root relative squared error (RRSE). The values of these parameters will determine how well the model performed.

3.2.1 R^2 Values of Models Used for Prediction of Mechanical Properties of Earth-Based Composites

The R^2, which is the coefficient of determination is the measure of the goodness of fit of a model. R-squared (R^2) is a statistical measure that gives an indication of how well the model regression line fit into the actual data [30]. It gives an indication of the strength of relationship between the linear model and the dependent variables. R-squared value ranges from 0 to 1 or 0 to 100%. High value of R^2 is an indication that the model have performed well in predicting the outcomes and vice versa. The Comparison of the performances of different models for prediction of the mechanical properties of earth-based composites using R^2 is presented in Table 3. Random Surface Methodology, Artificial Neural Networks and Support Vector Machine with R^2 value of 99% gave the best outcome in predicting split tensile strength, flexural strength and compressive strength of earth-based materials.

3.2.2 Root Mean Square Error (RMSE) Values of Models Used for Prediction of Mechanical Properties of Earth-Based Composites

Root mean square error is one of the performance indicator used in regression analysis for measuring the average of the difference between the model's predicted values and actual values. Lower value of root mean square error parameter indicates low error and better predicted output results [47]. Comparison of the root mean square error (RMSE) Values of models used for prediction of the mechanical properties of earth-based composites is presented in Table 4. Artificial Neural Networks having RMSE value of 0.04 MPa (0.15 MPa) outperformed other models in predicting the flexural and tensile strength (compressive strength) of earth-based composites while multivariate adaptive regression splines (Piecewise linear model) with outperformed other models in predicting the California bearing ratio of earth-based composites.

3.2.3 Mean Absolute Error (MAE) Values of Models Used for Prediction of Mechanical Properties of Earth-Based Composites

Mean absolute error (MAE) is used to assess the perfomance of regression models in machine learning and it is obtained by measuring the errors between paired observations in the same dataset. Comparison of the mean absolute error (MAE) values of models used for prediction of the mechanical properties of earth-based composites is presented in Table 5. The models with low values of mean absolute error predict better outcomes when compared with models with higher values of mean absolute error. Artificial Neural Networks having MAE value of 0.03 MPa outperformed other models in predicting splitting tensile and flexural strength while Multivariate adaptive regression splines (Piecewise linear model) having MAE value of 0.02 MPa

Table 3 Comparison of the performances of different models for earth-based composites using R^2

References	Model name	Output variable	R^2 (%)
[31]	Random Surface Methodology	Split tensile strength	99
[26]	Artificial Neural Networks	Flexural strength	99
[32]	Artificial Neural Networks	Compressive strength	99
[33]	Artificial Neural Networks	Compressive strength	99
[33]	Artificial Neural Networks	Splitting tensile strength	99
[33]	Artificial Neural Networks	Flexural and tensile strength	99
[28]	Support Vector Machine	Compressive strength	99
[28]	Artificial Neural Networks	Compressive strength	99
[20]	Artificial Neural Networks	Unit weight of soil	99
[31]	Random Surface Methodology	Flexural strength	98
[8]	Multivariate Regression	Compressive strength	97
[27]	Gradient Boosting Decision Tree	Splitting tensile strength	97
[34]	Multivariate adaptive regression splines (Piecewise linear model)	California bearing ratio	97
[35]	Extreme learning machine-cooperation Search optimizer	California bearing ratio	97
[13]	Radial basis function	Maximum dry density	97
[27]	Gradient Boosting Decision Tree	Splitting tensile strength	97
[9]	Multivariate Regression Model	Flexural strength	96
[36]	Random Forest	Compressive strength	96
[37]	Random Surface Methodology	Splitting tensile strength	94
[38]	Multi-expression programming	Compressive strength	91
[39]	Equivalent Flexural Strength Predictive model	Flexural strength	90
[40]	Gradient boosting (GB)	Unconfined compressive strength	90
[41]	SVM Regressor	Flexural strength	90
[42]	Random Forest	Cohesion	90
[43]	Random Forest	California bearing ratio	84
[44]	Decision Tree	Compressive strength	81
[25]	Gradient Boosting Decision Tree	Tensile strength	79
[31]	Random Surface Methodology	Compressive strength	74
[37]	Random Surface Methodology	Flexural strength	71
[43]	Artificial Neural Networks	California bearing ratio	67
[43]	Decision Tree	California bearing ratio	66
[45]	Artificial Neural Networks	Ultimate compressive strength	65
[45]	Artificial Neural Networks	Optimum moisture content	55
[43]	Linear Regression	California bearing ratio	53

(continued)

Table 3 (continued)

References	Model name	Output variable	R^2 (%)
[45]	Artificial Neural Networks	Maximum dry density	25
[46]	Geographically Weighted Regression	Soil quality	21

Table 4 Comparison of the performances of different models using root mean square error

References	Model name	Output variable	RMSE (MPa)
[33]	Artificial Neural Networks	Splitting tensile strength	0.04
[33]	Artificial Neural Networks	Flexural and tensile strength	0.04
[34]	Multivariate adaptive regression splines (Piecewise linear model)	California bearing ratio	0.04
[32]	Artificial Neural Networks	Compressive strength	0.15
[46]	Geographically Weighted Regression	Soil quality	0.15
[33]	Artificial Neural Networks	Compressive strength	0.32
[35]	Extreme learning machine-cooperation Search optimizer	California bearing ratio	0.48
[36]	Random Forest model	Compressive strength	3.04
[28]	Support Vector Machine	Compressive strength	3.56
[48]	Gaussian Process Regression (GPR) model	Tensile strength	5.327
[28]	Artificial Neural Networks	Compressive strength	6.33
[43]	Random Forest	California bearing ratio	8.13
[43]	Artificial Neural Networks	California bearing ratio	10.36
[43]	Linear Regression	California bearing ratio	11.08
[43]	Decision Tree	California bearing ratio	11.92

outperformed other models in predicting California bearing ratio of earth-based composites.

3.2.4 Coefficient of Correlation (CC) Values of Various Models Used for Prediction of Mechanical Properties of Prediction of Mechanical Properties of Earth-Based Composites

Coefficient of correlation is a statistical measure that indicates the strength of linear correlation/relationship between two variables. Coefficient of correlation values ranges from -1 to $+1$. Lower values of the coefficient of correlation indicates poor performance of the model's predicted outcomes whereas higher values of coefficient

Table 5 Comparison of the performance of different models using mean absolute error

References	Model name	Output variable	MAE (MPa)
[34]	Multivariate adaptive regression splines (Piecewise linear model)	California bearing ratio	0.02
[33]	Artificial Neural Networks	Splitting tensile strength	0.03
[33]	Artificial Neural Networks	Flexural and tensile strength	0.03
[33]	Artificial Neural Networks	Compressive strength	0.26
[36]	Random Forest model	Compressive strength	2.23
[28]	Support Vector Machine	Compressive strength	3.02
[48]	Gaussian Process Regression (GPR) model	Tensile strength	3.54
[43]	Random Forest	California bearing ratio	4.80
[28]	Artificial Neural Networks	Compressive strength	5.19
[43]	Decision Tree	California bearing ratio	5.30
[43]	Artificial Neural Networks	California bearing ratio	5.69
[43]	Linear Regression	California bearing ratio	8.90

Table 6 Comparison of the performances of different models using coefficient of correlation

References	Model name	Output variable	CC
[49]	Artificial Neural Networks	Compressive strength	99
[49]	Artificial Neural Networks	Tensile strength	98
[46]	Geographically Weighted Regression	Soil quality	39

of correlation depicts better performance of the model's predicted outcomes. Table 6 shows the Comparison of the Coefficient of correlation (CC) values of various models used for prediction of the mechanical properties of earth-based composites. Artificial Neural Networks having CC value of 99% outperformed Geographically Weighted Regression in predicting the properties of earth-based composites.

3.2.5 Relative Absolute Error (RAE) Values of Models Used for Prediction of Mechanical Properties of Earth-Based Composites

Relative absolute error is obtained by taking the ratio of the absolute error of the measurement to the actual measurement. The comparison of the relative absolute error (RAE) values obtained from models used for prediction of the mechanical properties of earth-based composites is presented in Table 7. Models' result outputs with lower values of relative absolute error indicates better performance. Back Propagation Neural Network having RAE value of 7.16% outperformed Artificial Neural Networks for prediction of compressive strength of earth-based composites.

Table 7 Comparison of the performances of different models using relative absolute error

References	Model name	Output variable	RAE (%)
[33]	Artificial Neural Networks	Splitting tensile strength	6.15
[33]	Artificial Neural Networks	Flexural tensile strength	7.13
[50]	Back Propagation Neural Network	Compressive strength	7.16
[33]	Artificial Neural Networks	Compressive strength	10.47

Table 8 Comparison of the performances of different models using root relative squared error

References	Model name	Output Variable	RRSE (%)
[33]	Artificial Neural Networks	Splitting tensile strength	6.50
[33]	Artificial Neural Networks	Flexural tensile strength	7.64
[33]	Artificial Neural Networks	Compressive strength	9.30

3.2.6 Root Relative Squared Error (RRSE) Values of Models Used for Prediction of the Mechanical Properties of Earth-Based Composites

Root relative squared error is used to normalize the total squared error by dividing it by the total squared error of the simple predictor. The Comparison of the root relative squared error (RRSE) values of models used for prediction of the mechanical properties of earth-based composites is presented in Table 8. Lower values of root relative squared error implies that the model used predicts better outcome.

4 Conclusions

In this chapter, machine learning models for predicting the mechanical properties of earth-based composites for sustainable building applications were evaluated. There has been increasing number of research in the application of soft computing in Civil Engineering. Several predictive models have been explored by different researchers. Among these predictive models, some have shown superior performance than others. Only predictive machine learning models that have been explored by researchers for predicting the mechanical properties of earth-based composites are covered in this chapter. The performance of the identified machine learning models for predicting the mechanical properties of earth-based composites were evaluated with respect to the R-squared values, root mean square error (RMSE), coefficient of correlation (CC), root absolute error (RAE), mean absolute error (MAE) and root relative squared error (RRSE).

In the area of earth-based composites, new problems on optimization of strength and durability of earth-based composites using machine learning models are not

widely investigated yet. Therefore, there is great need for increasing trend on optimization of strength and durability of earth-based composites by utilizing high performance and innovative machine learning models.

References

1. Onyegiri, I., Ben Ugochukwu, I.: Traditional building materials as a sustainable resource and material for low cost housing in Nigeria: advantages, challenges and the way forward. Int. J. Res. Chem. Metall. Civ. Eng. **3**(2) (Aug 2016). https://doi.org/10.15242/ijrcmce.u0716311
2. Khan, M., McNally, C.: A holistic review on the contribution of civil engineers for driving sustainable concrete construction in the built environment. Dev. Built Environ. **16** (01 Dec 2023). Elsevier Ltd. https://doi.org/10.1016/j.dibe.2023.100273
3. Taiwo, A., Adeboye, A.: Sustainable Housing Supply in Nigeria Through the Use of Indigenous and Composite Building Materials (2013). www.iiste.org
4. Obianyo, I.I., et al.: Production and utilization of earth-based bricks for sustainable building applications in Nigeria: status, benefits, challenges and way forward. J. Build. Pathol. Rehabil. **6**(1) (Dec 2021). https://doi.org/10.1007/s41024-021-00131-4
5. Onyelowe, K.C., Obianyo, I.I., Onwualu, A.P., Onyia, M.E., Moses, C.: Morphology and mineralogy of rice husk ash treated soil for green and sustainable landfill liner construction. Clean. Mater. **1** (Dec 2021). https://doi.org/10.1016/j.clema.2021.100007
6. Obianyo, I.I., Onwualu, A.P., Soboyejo, A.B.O.: Mechanical behaviour of lateritic soil stabilized with bone ash and hydrated lime for sustainable building applications. Case Stud. Constr. Mater. **12** (Jun 2020). https://doi.org/10.1016/j.cscm.2020.e00331
7. Onyelowe, K.C., Obianyo, I.I.: Influence of moisture and geofluids (GF) on the morphology of quarry fines treated lateritic soil. Clean. Eng. Technol. **3** (Jul 2021). https://doi.org/10.1016/j.clet.2021.100120
8. Obianyo, I.I., et al.: Multivariate regression models for predicting the compressive strength of bone ash stabilized lateritic soil for sustainable building. Constr. Build. Mater. **263** (Dec 2020). https://doi.org/10.1016/j.conbuildmat.2020.120677
9. Stanislas, T.T., et al.: Multivariate regression approaches to predict the flexural performance of cellulose fibre reinforced extruded earth bricks for sustainable buildings. Clean. Mater. **7** (Mar 2023). https://doi.org/10.1016/j.clema.2023.100180
10. Mahamat, A.A., et al.: Machine learning approaches for prediction of the compressive strength of alkali activated termite mound soil. Appl. Sci. (Switzerland) **11**(11) (2021). https://doi.org/10.3390/app11114754
11. Huang, J., Zhou, M., Yuan, H., Sabri, M.M.S., Li, X.: Towards sustainable construction materials: a comparative study of prediction models for green concrete with metakaolin. Buildings **12**(6) (Jun 2022). https://doi.org/10.3390/buildings12060772
12. Alabi, S.A., Mahachi, J.: Predictive models for evaluation of compressive and split tensile strengths of recycled aggregate concrete containing lathe waste steel fiber. Jordan J. Civ. Eng. **14**(4), 598–607 (2020)
13. Alavi, A.H., Gandomi, A.H., Gandomi, M., Sadat Hosseini, S.S.: Prediction of maximum dry density and optimum moisture content of stabilised soil using RBF neural networks. IES J. Part A: Civ. Struct. Eng. **2**(2), 98–106 (2009). https://doi.org/10.1080/19373260802659226
14. Kelechi, S.E., et al.: Modeling and optimization of high-volume fly ash self-compacting concrete containing crumb rubber and calcium carbide residue using response surface methodology. Arab. J. Sci. Eng. **47**(10), 13467–13486 (2022). https://doi.org/10.1007/s13369-022-06850-1
15. Kate, S., Swami, V., phode, S., Singh, V.: Advanced applications of artificial ıntelligent systems in civil engineering: a review. In: IOP Conference Series: Earth and Environmental Science. IOP Publishing Ltd. (Aug 2021).https://doi.org/10.1088/1755-1315/822/1/012009

16. Jia, J., Li, Y.: Deep learning for structural health monitoring: data, algorithms, applications, challenges, and trends. Sensors (Basel, Switzerland) **23**(21) (30 Oct 2023). https://doi.org/10.3390/s23218824
17. Malekloo, A., Ozer, E., AlHamaydeh, M., Girolami, M.: Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. Struct. Health Monit. **21**(4), 1906–1955 (2022). Sage Publications Ltd. https://doi.org/10.1177/14759217211036880
18. Rodrigues, M., Miguéis, V.L., Felix, C., Rodrigues, C.: Machine learning and cointegration for structural health monitoring of a model under environmental effects. Expert. Syst. Appl. **238** (Mar 2024). https://doi.org/10.1016/j.eswa.2023.121739
19. Sathiparan, N., Jeyananthan, P.: Soft computing techniques to predict the compressive strength of groundnut shell ash-blended concrete. J. Eng. Appl. Sci. **70**(1) (Dec 2023). https://doi.org/10.1186/s44147-023-00302-9
20. Straz, G., Borowiec, A.: Estimating the unitweight of local organic soils from laboratory tests using artificial neural networks. Appl. Sci. (Switzerland) **10**(7) (Apr 2020), https://doi.org/10.3390/app10072261
21. Datta, S.D., Islam, M., Rahman Sobuz, M.H., Ahmed, S., Kar, M.: Artificial intelligence and machine learning applications in the project lifecycle of the construction industry: a comprehensive review. Heliyon **10**(5) (15 Mar 2024). Elsevier Ltd. https://doi.org/10.1016/j.heliyon.2024.e26888
22. Sun, H., Burton, H.V., Asce, M., Huang, H.: Machine learning applications for building structural design and performance assessment: 1 state-of-the-art review 2 (2020)
23. Shao, W., et al.: The application of machine learning techniques in geotechnical engineering: a review and comparison. Mathematics **11**(18) (01 Sep 2023). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/math11183976
24. Vadyala, S. R., Betgeri, S.N., Matthews, J.C., Matthews, E.: A review of physics-based machine learning in civil engineering (2022)
25. Wang, W., Wu, Y., Liu, W., Fu, T., Qiu, R., Wu, S.: Tensile performance mechanism for bamboo fiber-reinforced, palm oil-based resin bio-composites using finite element simulation and machine learning. Polymers (Basel) **15**(12) (Jun 2023). https://doi.org/10.3390/polym15122633
26. Wang, X.Y.: Prediction of flexural strength of natural pozzolana and limestone blended concrete using machine learning based models. In: IOP Conference Series: Materials Science and Engineering. Institute of Physics Publishing (Apr 2020).https://doi.org/10.1088/1757-899X/784/1/012005
27. Li, Q., et al.: Splitting tensile strength prediction of Metakaolin concrete using machine learning techniques. Sci. Rep. **13**(1) (Dec 2023). https://doi.org/10.1038/s41598-023-47196-4
28. Chen, H., Qian, C., Liang, C., Kang, W.: An approach for predicting the compressive strength of cement-based materials exposed to sulfate attack. PLoS One **13**(1) (Jan 2018). https://doi.org/10.1371/journal.pone.0191370
29. Iftikhar, B., et al.: A machine learning-based genetic programming approach for the sustainable production of plastic sand paver blocks. J. Market. Res. **25**, 5705–5719 (2023). https://doi.org/10.1016/j.jmrt.2023.07.034
30. Schober, P., Schwarte, L.A.: Correlation coefficients: appropriate use and interpretation. Anesth. Analg. **126**(5), 1763–1768 (2018). https://doi.org/10.1213/ANE.0000000000002864
31. Jae, I.A., et al.: Experimental and predictive evaluation of mechanical properties kenaf-polypropylene fibre reinforced concrete using response surface methodology. Dev. Built Environ., 100262 (Dec 2023). https://doi.org/10.1016/j.dibe.2023.100262
32. Alabi, S.A., Mahachi, J.: Predictive models for evaluation of compressive and split tensile strengths of recycled aggregate concrete containing lathe waste steel fiber
33. Upreti, K., et al.: Prediction of mechanical strength by using an artificial neural network and random forest algorithm. J. Nanomater. **2022** (2022). https://doi.org/10.1155/2022/7791582
34. Bardhan, A., Gokceoglu, C., Burman, A., Samui, P., Asteris, P.G.: Efficient computational techniques for predicting the California bearing ratio of soil in soaked conditions. Eng. Geol. **291** (Sep 2021). https://doi.org/10.1016/j.enggeo.2021.106239

35. Nagaraju, T.V., Bahrami, A., Prasad, C.D., Mantena, S., Biswal, M., Islam, M.R.: Predicting California bearing ratio of lateritic soils using hybrid machine learning technique. Buildings **13**(1) (Jan 2023). https://doi.org/10.3390/buildings13010255
36. Rahman Raju, M., Rahman, M., Mehedi Hasan, M., Sajib Hosen, M., Monirul Islam, M., Md Sadiqul Hasan, N.: A comparative study of machine learning methods for assessing the compressive strength of Pozzolanic concrete (2023). https://doi.org/10.21203/rs.3.rs-3288929/v1
37. Pinheiro, C., Rios, S., da Fonseca, V.A., Fernández-Jiménez, A., Cristelo, N.: Application of the response surface method to optimize alkali activated cements based on low-reactivity ladle furnace slag. Constr. Build. Mater. **264** (Dec 2020). https://doi.org/10.1016/j.conbuildmat.2020.120271
38. Iftikhar, B., et al.: Predicting compressive strength of eco-friendly plastic sand paver blocks using gene expression and artificial intelligence programming. Sci. Rep. **13**(1) (Dec 2023). https://doi.org/10.1038/s41598-023-39349-2
39. Donkor, P., Obonyo, E., Ferraro, C.: Fiber reinforced compressed earth blocks: evaluating flexural strength characteristics using short flexural beams. Materials **14**(22) (Nov 2021). https://doi.org/10.3390/ma14226906
40. Eyo, E.U., Abbey, S.J., Booth, C.A.: Strength predictive modelling of soils treated with calcium-based additives blended with eco-friendly pozzolans—a machine learning approach. Materials **15**(13) (Jul 2022). https://doi.org/10.3390/ma15134575
41. Ali, A., et al.: Machine learning-based predictive model for tensile and flexural strength of 3D-printed concrete. Materials **16**(11) (Jun 2023). https://doi.org/10.3390/ma16114149
42. Hoque, M.J., Bayezid, M., Sharan, A.R., Kabir, M.U., Tareque, T.: Prediction of strength properties of soft soil considering simple soil parameters. Open J. Civ. Eng. **13**(03), 479–496 (2023). https://doi.org/10.4236/ojce.2023.133035
43. Kassa, S.M., Wubineh, B.Z.: Use of machine learning to predict california bearing ratio of soils. Adv. Civ. Eng. **2023** (2023). https://doi.org/10.1155/2023/8198648
44. Anysz, H., Brzozowski, Ł., Kretowicz, W., Narloch, P.: Feature importance of stabilised rammed earth components affecting the compressive strength calculated with explainable artificial intelligence tools. Materials **13**(10) (May 2020). https://doi.org/10.3390/ma13102317
45. Taffese, W.Z., Abegaz, K.A.: Prediction of compaction and strength properties of amended soil using machine learning. Buildings **12**(5) (May 2022). https://doi.org/10.3390/buildings12050613
46. Isong, I.A., John, K., Okon, P.B., Ogban, P.I., Afu, S.M.: Soil quality estimation using environmental covariates and predictive models: an example from tropical soils of Nigeria. Ecol. Process. **11**(1) (Dec 2022). https://doi.org/10.1186/s13717-022-00411-y
47. Nau, R.: Review of basic statistics and the simplest forecasting model: the sample mean (2014)
48. Le, T.T.: Prediction of tensile strength of polymer carbon nanotube composites using practical machine learning method. J. Compos. Mater. **55**(6), 787–811 (2021). https://doi.org/10.1177/0021998320953540
49. Asteris, P.G., et al.: Masonry compressive strength prediction using artificial neural networks. In: Communications in Computer and Information Science, pp. 200–224. Springer Verlag (2019). https://doi.org/10.1007/978-3-030-12960-6_14
50. Turco, C., Funari, M.F., Teixeira, E., Mateus, R.: Artificial neural networks to predict the mechanical properties of natural fibre-reinforced compressed earth blocks (Cebs). Fibers **9**(12) (Dec 2021). https://doi.org/10.3390/fib9120078

Shear Wall Cost Optimization by Employing Harmony Search

Muhammed Çoşut, Gebrail Bekdaş, and Sinan Melih Nigdeli

Abstract In this study, shear walls that are employed for construction are optimized in terms of cost by using Harmony Search (HS) which is one of the metaheuristic algorithms. All necessary variables, constants, constraints along with equations are specified from the TS 500 (Requirements for Design and Construction of Reinforced Concrete Structures) [1] and TBDY 2018 (Turkiye Structures Earthquake Standard) [2] standards, which are employed to build a structure with an appropriate design and sufficient capacity under the loading. Regarding these standards, using shear walls is an indispensable structure element because they show good performance in case of earthquake and wind loading which are known as lateral loads. A building that is 3 floors and with a 9-m height is used, and all necessary unknowns are computed thanks to the algorithm in the program. C30/37 and C40/50 concrete classes are used separately, and objective functions' results are found to have different values because cross-section and reinforcement area affect the result of the objective function.

Keywords Shear wall · Standards · Metaheuristic algorithm · Harmony search · Cost optimization

1 Introduction

It is an undeniable fact that the pace rate of the development of engineering has been continuing by increasing throughout the recent decades. As a result, several functional and vertical systems are developed [3]; in addition, artificial intelligence has been enhanced to create convenient designs for many areas [4–7]. Different indispensable projects, as well as studies, have been completed successfully, which is a prominent process in order to build up and organize the life problems that individuals encounter

M. Çoşut · G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Avcılar, İstanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_10

not only as an everyday issue but also as a general lack of life standards. One of the crucial areas worldwide is known as Civil Engineering which consists of a variety of departments such as water management, materials, structures, mechanics and geotechnics. Looking at the mechanics and structures in more detail, there has been a lot of research to reach appropriate design in any area because these areas might have different characteristics, and design should be implemented according to the features of areas. Some countries encounter not only earthquakes but also wind loading which influences structures adversely; indeed, the construction in that area might be demolished by the destructive earthquake. Therefore, the pertinent design is vital to provide individuals with any hesitation. In order to protect structures, several structure elements which can be active control systems, passive control systems, hybrid control systems, brace elements and shear walls have been utilized to refrain from any damage in the construction [8]. As for shear walls, they are preferred in order to hinder any further damage such as unwanted displacements, and they ensure a structure displacement at a certain level, so lateral load resistance goes up regarding rigidity [9]. Furthermore, the usage of these walls is influenced by what kind of slabs are used in the system. The other important circumstance is the location of the shear walls according to the construction. They have to locate without changing too much to the centre of gravity, which prevents eccentricity in the system.

That a lot of studies have been accomplished in this area is important for future sustainable and pertinent design. Esmaili et al. have conducted the performance of shear walls in the building according to overall torsion, time-dependent effects and construction sequence loading, and the building was constructed in Tehran [10]. Data for machine learning has been collected from other studies' results by Zhang et al. [11] in order to attain acknowledgment of shear walls' capacity as regards strength, deformation capacity and forecasting of failure modes. In addition, Mangalathu et al. [12] have investigated a similar topic in their article on seismic failure identification by using data-driven machine learning. 86% accuracy rate of failure modes of the shear walls was found in this article based on the proposed method. Some shear walls have openings because of their functional reasons, and Kim and Lee [13] have controlled the technique's efficiency and accuracy by relying on static and dynamic loads, which results were found as satisfactory conclusions.

Although several optimization methods using metaheuristic algorithms have been developed for RC members [26] such as beams [27–29], columns [30–32], frames [33, 34], retaining walls [35] and footings [36], the optimum design of shear walls via metaheuristics has not been deeply investigated. In this study, shear walls' design which is significant for mostly seismic areas is carried out to achieve minimum cost value according to standards and loadings of the moment and axial force and concrete classes which are C30/37 and C40/50 by using a metaheuristic algorithm known as Harmony Search. In the design process, TS500 and TBDY 2018 standards are used to find the appropriate design.

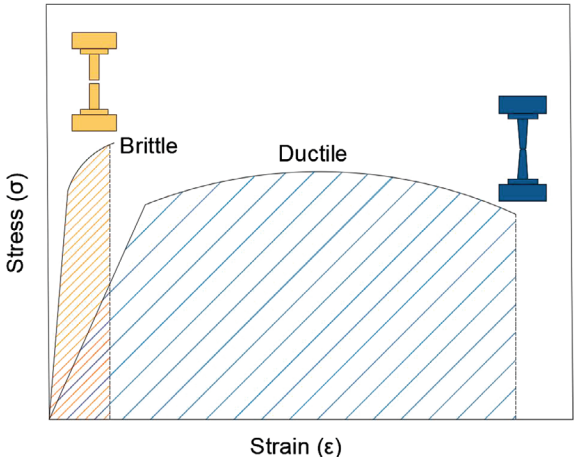


Fig. 1 Rigid and ductile elements' strain under the stress

2 Rigidity and Ductility

In order to design constructions, rigidity and ductility have an important role throughout the process because they directly impact the principle of operation of the system under the loads. As regards rigidity, it reduces the damage of destructive earthquakes. The location of the shear walls [14], joint connection [15], materials types along with cross-section dimensions affect the rigidity ratio.

Regarding ductility, it is that structural elements can displace at a certain level without decreasing their bearing capacity, and they have the ability to resist repetitive loadings by absorbing energy.

Figure 1 delineates that rigid elements can endure more stress, whereas ductile elements can displace more than rigid elements. As a consequence, there are powerful and poor features for these two members. Rigid members can provide structure to not displace or deteriorate the system's stability. In other words, under of the line for rigid and ductile elements show the area which is related to the capacity of energy absorption.

3 Shear Walls

The construction consists of several structural elements that have different features in the system. These can be beams, slabs, columns, shear walls, precast shear walls [16] along with control system members, which all have different properties when they are implemented. Beams contribute to the transfer of the loading to the column. Columns enable to convey all structure load to the foundation which can be designed of various types. Control system members are specified in terms of building height, affected load as well as desirable displacement.

When it comes to shear walls, these walls' behavior is determined by the material utilized, position, thickness and length of the wall. Earthquakes and wind loads which are lateral are endured by these walls, and also, they sometimes resist lateral earth pressure along with hydrostatic. Shear walls have high bending rigidity; thus, they contribute extra rigidity to the structures. They can also be implemented as shape and opening shear walls such as single-two rows along with staggered (Fig. 2). Shear walls can include doors or windows, in this case, loadings and reinforcement area should be computed for the opening zone. Furthermore, some conditions should be implemented such as stirrups' distance cannot be more than 150 mm [17], and extra rebar should be added.

All floors have different displacements, so they should ratio with each other, and if this ratio is more than the usual one, torsional irregularity appears in the system, which is an unwanted circumstance. Therefore, shear walls should be located appropriately to increase the torsional rigidity. Moreover, if the plan is not designed as well and balanced, some shear walls might expose much more loads compared to other shear walls, which ruins the stability of the system, and maybe the system will be collapsed by seismic loads. In the process of design, the gravity centre and rigidity centre should be the same coordinated (Fig. 3). %5 eccentrics between these centres are allowed by standards.

Figure 3 displays the examples of shear wall locations, and the second and third ones have more good places than the first ones which also have an appropriate design. Two coordinate systems which are the x-axis, as well as the y-axis, should be generated as balance. As a consequence, the performance of the building will be sufficient strength and stiffness not only according to the x-axis but also regarding the y-axis. Because, the earthquake direction is not known whether it's direction

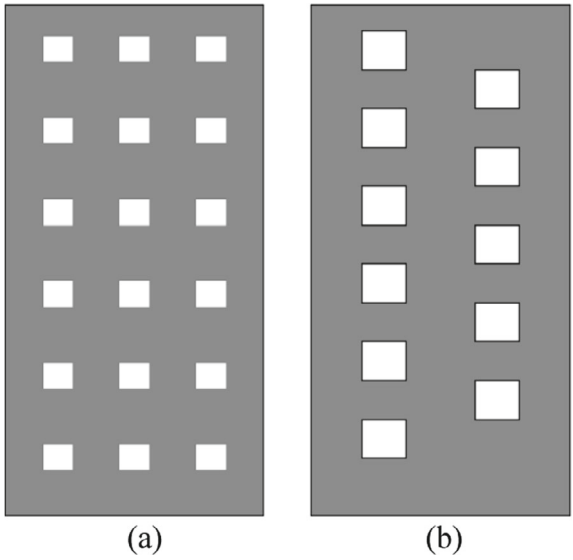


Fig. 2 Shear wall types. **a** Three rows, **b** Staggered

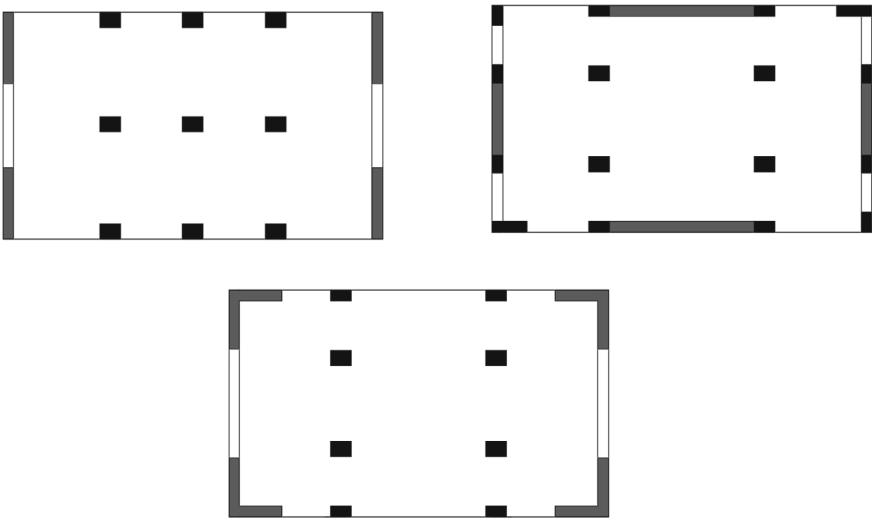


Fig. 3 Appropriate design

x-axis, y-axis, or z-axis, the distribution of shear walls in every direction allows the structure good performance under seismic loads. Therefore, the efficiency of cost and minimizing earthquake damage occur.

Figure 4 demonstrates that several improper shear walls design might be done in the past structures. These designs provide more drawbacks than benefits under loading.

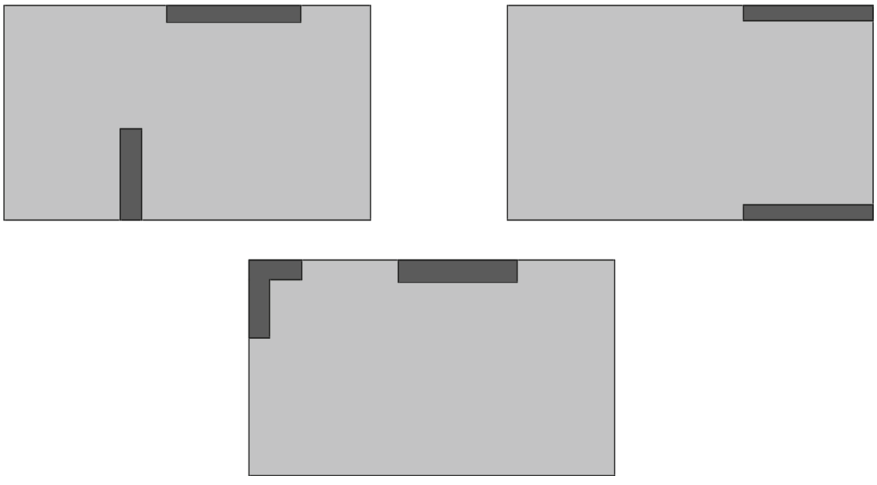


Fig. 4 Inappropriate design

Shear walls' plan has some important factors to take advantage of as high performance. These are:

- Shear walls generally should distribute in terms of the edge of plan and axis; thus they reduce the displacement of the system.
- Shear walls should carry most of the slab's loading.
- Earthquake loads should distribute relatively equal to shear walls.

Shear walls can bend in terms of seismic loading direction, which can be horizontal shear, vertical shear, buckling as well as flexural failure. Horizontal shear and vertical shear are delineated in Fig. 5. Every loading and failure situation has different results in the system; however, the fringe of shear walls is influenced more under bending moment.

There are some criteria for appropriate design for shear walls that are related to reinforcement areas. The reinforcement area plays an important role in designs to keep the structure safe and stable. Critical shear wall height should be computed, and then, the reinforcement areas for critical and non-critical areas are calculated with different equations. These equations are shown in Eqs. (1) and (2). b_w is the shear wall width, while l_w is the length of the shear wall.

$$Critical\ height = \begin{cases} 0.002 \times b_w \times l_w \\ 4\,\emptyset\,14 \end{cases} \tag{1}$$

$$Non{-}Critical\ height = \begin{cases} 0.001 \times b_w \times l_w \\ 4\,\emptyset\,14 \end{cases} \tag{2}$$

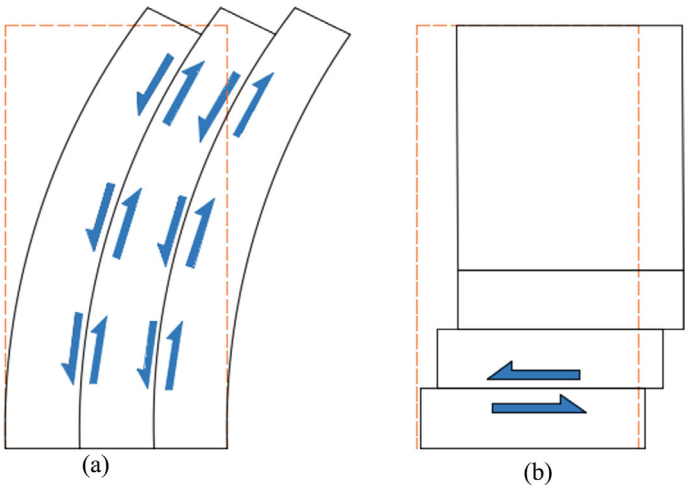


Fig. 5 **a** Vertical shear, **b** Horizontal shear

4 Metaheuristic Algorithm

Metaheuristic algorithms have been inspired by both nature and life events. This situation leads to several different properties algorithms which can be phase numbers or equations have emerged in a short period [18]. As a consequence, these features influence the time when the problem results will be reached. Some metaheuristic algorithms are Genetic Algorithm (GA) [19], Harmony Search (HS) [20], Jaya Algorithm (JA) [21], Shark Smell Optimization (SSO) [22], Bat Algorithm (BA) [23], Ant Colony Optimization (ACO) [24].

These algorithms are used to find objective functions in a short time. Objective functions can be varying such as minimizing cost, minimizing CO_2 emission, minimizing displacement, maximizing speed and maximizing machine capacity. Furthermore, the objective function can be generated as not only single-objective but also multi-objective, which enables the problem to be solved according to different features.

4.1 Harmony Search (HS)

One of the metaheuristic algorithms called Harmony Search which can be employed in several engineering studies, machine and energy systems [20] as well as finance was discovered by Geem et al. [25] in 2011. Song tone has inspired this algorithm to be found. The main purpose of this method is to discover unprecedented and great musical work. Therefore, the harmony of music and notation influences individuals and collects listeners' likes. In order to reach the top level, this process continues. On the other hand, musicians can choose improvisation to create a good artifact, and Fig. 6 demonstrates the options which can be preferred by musicians.

As a consequence, the objective function can be defined as unprecedented music, and all attempts to reach good quality music have similar features with maximum iteration numbers. In addition, Harmony Search consists of some unique equations which

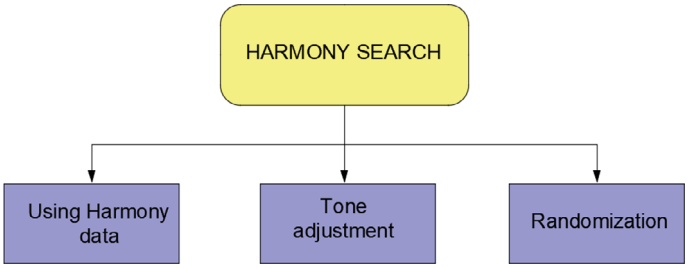


Fig. 6 Harmony searches for diversity decisions

include HCMR and PAR. HCMR which is known as Harmony Memory Consideration Rate changes between 0 and 1, and PAR is known as Pitch Adjustment Rate.

Equation (3) demonstrates the Harmony Search equations. The maximum value of the i. the design parameter is $X_{i,\max}$, while the minimum value of the i. the design parameter is $X_{i,\min}$. Moreover, $X_{i,k}$: The value of the candidate solution i. and j. in the initial matrix, $rand(-\frac{1}{2}, \frac{1}{2})$ is randomly specified between −0.5 and 0.5.

$$X'_{i,new} \begin{cases} HCMR > rand(), X_{i,\min} + rand() \times \left(X_{i,\max} - X_{i,\min}\right) \\ HCMR < rand(), X_{i,k} + rand\left(-\frac{1}{2}, \frac{1}{2}\right) \times PAR \times \left(X_{i,\max} - X_{i,\min}\right) \end{cases} \tag{3}$$

5 Numerical Example

Table 1 displays the properties of cross-section and materials for designing shear walls. Standards are used to clarify all values, maximum-minimum cross sections, rebar yield strength, specific gravity for steel and cost of current materials.

Table 2 shows the necessary design values to compute shear walls in terms of cost; therefore, the objective function is chosen according to the usage of materials and their costs.

Figure 7 includes information on the calculation of shear walls, and abbreviations are defined in Table 2. Additionally, H_{cr} is critical shear wall height, l_{u1} and l_{u2} provide the display for longitudinal reinforcement placement.

Constraints for shear walls are demonstrated in Table 3 which include various equations. To exemplify, constraints are height length ratio, critical shear walls height, longitudinal and width reinforcement, shear resistance along with capacity designs.

In addition, there should be the calculation of shear walls with high ductility as well as shear walls with normal ductility.

Table 1 Cross-section and materials features

Explanation	Symbol	Unit	Value or formula
Minimum section width	$b_{w,min}$	mm	200
Maximum section width	$b_{w,max}$	mm	400
Minimum section length	$l_{w,min}$	mm	1500
Maximum section length	$l_{w,max}$	mm	2500
Yield strength of concrete	f_{yk}	MPa	420
The specific gravity of steel	γ_s	t/m^3	7.86
Cost of concrete per unit volume	C_c	TL/m^3	C30/37 → 1700 C40/50 → 1950
Cost of steel per unit weight	C_s	TL/ton	19,350

Table 2 Design values

Explanation	Symbol	Unit	Value
Storey	–	–	3
Shear wall height	H_w	m	9
Coefficient of excess strength	D	–	2.5
Maximum section length	Vt	kN	1000
Design shear force	V_d	kN	442
Design normal force	N_d	kN	825
Design moment force	M_d	kN m	1500
The sum of elements cross-sectional area	A_g	m^2	1
The sum of the plan areas of the floors	A_p	m^2	300
Clear cover	f_{yk}	MPa	30

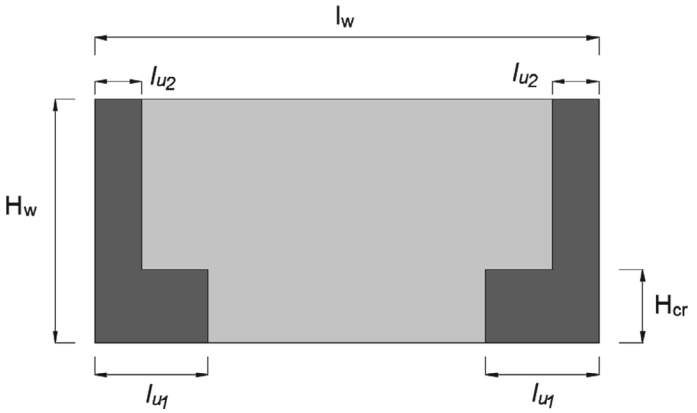


Fig. 7 Shear wall

Table 3 Constraints for design

Explanation	Limits
Height length ratio	$\frac{H_w}{l_w} > 2$
Length width ratio	$\frac{l_w}{b_w} \geq 6$
Critical shear walls height	Critical height (H_{cr}) $\leq 2 \times l_w$
Area ratio	$\frac{\sum A_p}{\sum A_g}$
Shear force ratio	$\frac{V_t}{\sum A_g} \leq 0.5 \times f_{ctd}$

With regards to given properties such as variables, constants as well as constraints, they all have been used in the algorithm to reach objective functions based on standards. Table 4 displays the results of the C30/37 concrete class, and it includes cross-sections, reinforcement area cost, concrete cost and minimum cost which is given as TL (Turkish Lira). According to the cross-section and reinforcement area, the objective function was found at 25,786 TL.

Constraints' limits for C30/37 concrete class are shown in Table 5 which include height length ratio, length–width ratio, critical shear wall height, area ratio and shear force ratio. When all limit values are thoroughly checked, no discrepancies are observed, indicating that the design has been accurately executed. In fact, if any value had not complied, the objective function would have been penalized, and those cross-sectional values would not have been deemed suitable for the system.

Moreover, comparisons between shear load on high ductile shear walls and normal ductile shear walls results were computed and checked whether there was any conflict between these results, and the design seems appropriate in terms of loading and features of materials.

To show different concrete class results, C40/50 concrete class was chosen and utilized in the algorithm, and all constraints, variables and constants are taken as C30/37 concrete class values. Table 6 provides acknowledgment of the shear wall's cross-section, critical length, the critical height as well as costs for C40/50 concrete class. The minimum cost to design shear walls is calculated at 23,809 TL.

Table 4 C30/37 concrete class results

Symbols	Units	Results
b_w	mm	358.52
l_w	mm	2182.5
l_{u1}	mm	717
l_{u2}	mm	358.5
H_{cr}	mm	2182.5
Concrete cost	TL	11,972
Rebar cost	TL	13,815
Objective function	TL	25,786

Table 5 Constraints' limits result for C30/37

Explanation	Results	Limits
Height length ratio	4.12	$\frac{H_w}{l_w} > 2$
Length width ratio	6.08	$\frac{l_w}{b_w} \geq 6$
Critical shear walls height	4365	Critical height (H_{cr}) $\leq 2 \times l_w$
Area ratio	0.0052	$\frac{\sum A_p}{\sum A_g}$
Shear force ratio	639	$\frac{V_t}{\sum A_g} \leq 0.5 \times f_{ctd}$

Table 6 C40/50 concrete class results

Symbols	Units	Results
b_w	mm	334.2
l_w	mm	2028
l_{u1}	mm	668.3
l_{u2}	mm	334.15
H_{cr}	mm	2028.1
Concrete cost	TL	11,893
Rebar cost	TL	11,916
Objective function	TL	23,809

Table 7 Constraints' limits result for C40/50

Explanation	Results	Limits
Height length ratio	4.44	$\frac{H_w}{l_w} > 2$
Length width ratio	6.07	$\frac{l_w}{b_w} \geq 6$
Critical shear walls height	4056	Critical height (H_{cr}) $\leq 2 \times l_w$
Area ratio	0.0045	$\frac{\sum A_p}{\sum A_g}$
Shear force ratio	737.85	$\frac{V_t}{\sum A_g} \leq 0.5 \times f_{ctd}$

Table 7 gives information about the constraint's limits result for C40/50 concrete class. It is seen that the values found provide the constraint values.

To delineate the cost differences between concrete class, concrete cost, steel cost and total cost are given in Fig. 8. Concrete costs are slightly the same, whereas steel costs have a considerable difference, at approximately 2000 TL. Due to the steel cost difference, the total costs for concrete classes were found to be different amounts, and C40/50 concrete class has a cheaper design compared to C30/37 concrete class design.

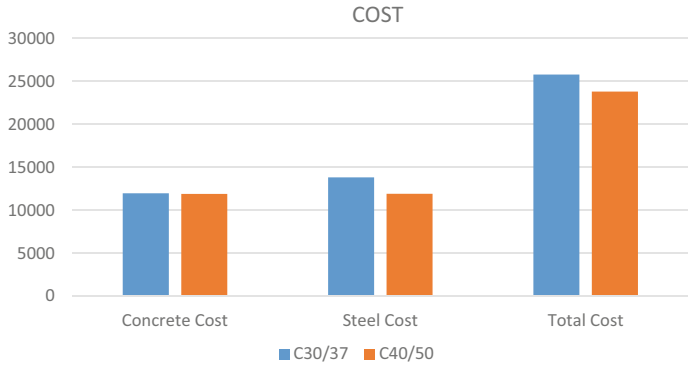


Fig. 8 Comparison between C30/37 and C40/50 cost results

6 Conclusion

Shear walls are frequently used in construction to create suitable building systems against lateral loads. Due to the significance of their usage within the structural system, numerous laboratory experiments and numerical studies have been conducted to establish standards. These standards encompass various factors that determine the width, length, reinforcement area, and constraint values of the structural elements. This enables the positioning of sections in earthquake-prone regions based on soil class and seismicity to ensure a design that complies with appropriate criteria. However, if the placement of shear walls significantly increases the distance between the centre of gravity and the mass centre, structural irregularities may occur, leading to undesirable structural deficiencies. Hence, it is crucial to pay attention to the equations and constraints specified in the standards during the design phase.

Considering the TS500 and TBDY 2018 standards, design equations were implemented in MATLAB, and the Harmony Search algorithm was utilized to achieve the objective function of minimizing the cost. Constraint values were penalized with a high cost if they did not comply with the desired standard values. This approach led to the attainment of a minimum-cost design in accordance with the standards. The objective function was separately calculated for both C30/37 and C40/50 concrete classes. For the given shear wall characteristics, the minimum cost was found to be 25,786 TL when using C30/37 concrete and 23,809 TL when using C40/50 concrete, at approximately 11% cost differences. The reason for C40/50 concrete being more cost-effective is due to the utilization of the design concrete value in the calculations, allowing for elements with less reinforcement and cross-section compared to C30/37 concrete. The cost of reinforcement being more than 8 times the cost of concrete reduces the overall cost. To obtain more cost-effective elements, the use of high-strength concrete may be considered, but it is crucial to be mindful that as concrete strength increases, it becomes more brittle, which is an undesirable property

for structural elements. Furthermore, standards have established limit values to keep the concrete grade within specific levels.

References

1. TS 500 (Betonarme Yapilarin Tasarim Ve Yapim Kurallari) (Requirements for Design and Construction of Reinforced Concrete Structures). Türk Standardlari Enstitüsü, Ankara, Turkey, (2000)
2. TBDY (Türkiye Bina Deprem Yönetmeliği), Afet ve Acil Durum Yönetimi Başkanlığı, Ankara, Turkey (2018)
3. Oliva, M., Barone, G., Navarra, G.: Optimal design of Nonlinear Energy Sinks for SDOF structures subjected to white noise base excitations. Eng. Struct. **145**, 135–152 (2017)
4. Çoşut, M., Bekdaş, G., Niğdeli, S.M.: Optimization of reinforced concrete frame structures and matrix displacement method. Chall. J. Concr. Res. Lett. **14**(1), 10–17 (2023)
5. Perea, C., Alcala, J., Yepes, V., Gonzalez-Vidosa, F., Hospitaler, A.: Design of reinforced concrete bridge frames by heuristic optimization. Adv. Eng. Softw. **39**(8), 676–688 (2008)
6. Jones, M.T.: Artificial intelligence application programming. Charles River Media, Hingham (Massachussets), USA (2003)
7. Sriram, R.D.: Artificial intelligence in engineering: personal reflections. Adv. Eng. Informat **20**, 3–5 (2006)
8. Çoşut, M., Nigdeli, S.M., Bekdaş, G.: Structural control systems and tuned mass damper optimization by using Jaya and hybrid algorithms. In: Hybrid Metaheuristics in Structural Engineering: Including Machine Learning Applications, pp. 87–110. Springer Nature Switzerland, Cham (2023)
9. Harne, V.R.: Comparative study of strength of RC shear wall at different location on Multistoried residential building. Int. J. Civ. Eng. Res. **5**(4), 391–400 (2014)
10. Esmaili, O., Epackachi, S., Samadzad, M., Mirghaderi, S.R.: Study of structural RC shear wall system in a 56-story RC tall building. In: The 14th World Conference Earthquake Engineering (2008)
11. Zhang, H., Cheng, X., Li, Y., Du, X.: Prediction of failure modes, strength, and deformation capacity of RC shear walls through machine learning. J. Build. Eng. **50**, 104145 (2022)
12. Mangalathu, S., Jang, H., Hwang, S.H., Jeon, J.S.: Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls. Eng. Struct. **208**, 110331 (2020)
13. Kim, H.S., Lee, D.G.: Analysis of shear wall with openings using super elements. Eng. Struct. **25**(8), 981–991 (2003)
14. Singhal, S., Chourasia, A., Chellappa, S., Parashar, J.: Precast reinforced concrete shear walls: State of the art review. Struct. Concr. **20**(3), 886–898 (2019)
15. Sørensen, J.H., Hoang, L.C., Olesen, J.F., Fischer, G.: Test and analysis of a new ductile shear connection design for RC shear walls. Struct. Concr. **18**(1), 189–204 (2017)
16. Wu, S., Li, H., Wang, X., Li, R., Tian, C., Hou, Q.: Seismic performance of a novel partial precast RC shear wall with reserved cast-in-place base and wall edges. Soil Dyn. Earthq. Eng. **152**, 107038 (2022)
17. Doğangün A.: Betonarme Yapıların Hesap ve Tasarımı, Birsen Yayın ve Dağıtım, İstanbul (2019)
18. Bekdaş, G., Nigdeli, M.N., Yücel, M., Kayabekir, A.E.: Yapay Zeka Optimizasyon Algoritmaları ve Mühendislik Uygulamaları, Seçkin Yayıncılık, Ankara (2021)
19. Frohlich, H., Chapelle, O., Scholkopf, B.: Feature selection for support vector machines by means of genetic algorithm. In: 15th IEEE International Conference on Tools with Artificial Intelligence, pp. 142–148 (2003)
20. Gao, X.Z., Govindasamy, V., Xu, H., Wang, X., Zenger, K.: Harmony search method: theory and applications. Comput. Intell. Neurosci., 39 (2015)

21. Rao, R.V.: Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. **7**, 19–34 (2016)
22. Abedinia, O., Amjady, N., Ghasemi, A.: A new metaheuristic algorithm based on shark smell optimization. Complexity **21**(5), 97–116 (2016)
23. Yang, X.S., He, X.: Bat algorithm: literature review and applications. Int. J. Bio-Inspired Comput. **5**(3), 141–149 (2013)
24. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, London (2004)
25. Geem, Z.W., Geem, E.: Music-Inspired Harmony Search Algorithm. Springer, Berlin, Germany (2009)
26. Kayabekir, A.E., Bekdaş, G., Nigdeli, S.M.: Metaheuristic approaches for optimum design of reinforced concrete structures: emerging research and opportunities: emerging research and opportunities (2020)
27. Rath, D.P., Ahlawat, A.S., Ramaswamy, A.: Shape optimization of RC flexural members. J. Struct. Eng. **125**(12), 1439–1446 (1999)
28. Jahjouh, M.M., Arafa, M.H., Alqedra, M.A.: Artificial Bee Colony (ABC) algorithm in the design optimization of RC continuous beams. Struct. Multidiscip. Optim. **47**, 963–979 (2013)
29. Azam, R., Riaz, M.R., Farooq, M.U., Ali, F., Mohsan, M., Deifalla, A.F., Mohamed, A.M.: Optimization-based economical flexural design of singly reinforced concrete beams: a parametric study. Materials **15**(9), 3223 (2022)
30. Bekdas, G., Nigdeli, S.M.: The optimization of slender reinforced concrete columns. PAMM **14**(1), 183–184 (2014)
31. de Medeiros, G.F., Kripka, M.: Optimization of reinforced concrete columns according to different environmental impact assessment parameters. Eng. Struct. **59**, 185–194 (2014)
32. Nigdeli, S.M., Bekdas, G., Kim, S., Geem, Z.W.: A novel harmony search based optimization of reinforced concrete biaxially loaded columns. Struct. Eng. Mech.: Int. J. **54**(6), 1097–1109 (2015)
33. Bekdaş, G., Nigdeli, S.M., Kim, S., Geem, Z.W.: Modified harmony search algorithm-based optimization for eco-friendly reinforced concrete frames. Sustainability **14**(6), 3361 (2022)
34. Paya-Zaforteza, I., Yepes, V., Hospitaler, A., Gonzalez-Vidosa, F.: CO_2-optimization of reinforced concrete frames by simulated annealing. Eng. Struct. **31**(7), 1501–1508 (2009)
35. Kayabekir, A.E., Arama, Z.A., Bekdaş, G., Nigdeli, S.M., Geem, Z.W.: Eco-friendly design of reinforced concrete retaining walls: multi-objective optimization with harmony search applications. Sustainability **12**(15), 6087 (2020)
36. Chaudhuri, P., Maity, D.: Cost optimization of rectangular RC footing using GA and UPSO. Soft. Comput. **24**, 709–721 (2020)

Effect of CatBoost Parameters on Cost Minimization of Rectangular Section Reinforced Concrete Columns Under Uniaxial Bending Effect

Yaren Aydın, Gebrail Bekdaş, and Sinan Melih Nigdeli

Abstract Reinforced concrete columns are vertical bearing elements that transfer loads from horizontal bearing elements such as slabs and beams to foundations. In this Chapter, the cost minimization of a rectangular reinforced concrete column under uniaxial bending is carried out and the obtained data is used in machine learning. Harmony Search (HS) was used for optimization and Categorical Boosting (CatBoost) was used for the machine learning process. The effect of model parameters on the success of the model is investigated using coefficient of determination (R^2). As a result, the "iterations" parameter had the highest impact on the model.

Keywords Column · HS · Cost optimization · Machine learning · Parameter effect · CatBoost

1 Introduction

Although concrete has high compressive strength, it is weak in terms of tensile strength. Reinforced concrete is the work of steel reinforcements, which have high tensile strength, working together in harmony by covering this weakness of concrete. In this way, a strong structure can be put forward in terms of mechanical properties.

Columns transmit the loads received from beams and slabs to the ground through foundations. Together with the beams, they form the frames, which are a good carrier system for the transportation of horizontal load effects. Since damage to the columns will weaken the entire structural system, care should be taken in their arrangement

Y. Aydın · G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University-Cerrahpaşa, Avcılar, 34320 Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

Y. Aydın
e-mail: yaren.aydin1@ogr.iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_11

and design [1]. Columns are the most important load-bearing elements in reinforced concrete frame structures. The duties of the columns in frame structures are to transfer all vertical and horizontal loads acting on the structure to the foundation system and to ensure that the relative storey drifts occurring under the influence of horizontal loads remain within the limits permitted by the regulations. Due to the monolithic characteristic of reinforced concrete frame systems, reinforced concrete columns carry bending moment and shear force as well as axial pressure.

Columns are the most important part of buildings and safety should be at the highest level in their design. All vertical loads in buildings are carried by columns. Most of the vertical loads are the weight of the reinforced concrete elements themselves. Since the axial forces acting on the column are large, bending moments generally have little effect on changing the direction of stresses. Since crushing of concrete is not accepted in reinforced concrete columns, axial force limitation, use of spiral reinforcement or stirrups in critical sections and consideration of second-order effects for columns are required to ensure ductility [2]. Minimum cost design of reinforced concrete columns, which constitute an important part of buildings, is important.

In this section, using harmony search (HS), a dataset is generated by performing the optimum design of the axially loaded column with bending moment. Then, using this dataset, the cost of rectangular reinforced concrete columns under uniaxial bending is estimated. The effect of changing the parameters of the machine learning (ML) model (CatBoost) used in cost prediction on the prediction success is investigated.

2 Material and Methods

2.1 Harmony Search (HS)

The Harmony Search Algorithm (HS) was developed in 2001 by Geem, Kim and Jonathan [3]. The Harmony Search algorithm aims to find the optimal result between the solutions it stores in its memory and the randomly generated solutions. This algorithm is inspired by the process of finding the best harmony in jazz music. In HS, the most aesthetic melody can be achieved when all orchestral elements work harmonically with each other. When this situation is adapted to optimization problems, the best solution can be obtained by approaching the global solution of the objective function in optimization problems [3].

In the literature, the HS algorithm is frequently used to solve many problems. Examples of these are university course timetabling [4], feature selection [5], design optimization of a stand-alone green energy system [6], fake news detection [7], privacy preservation in cloud data centers [8], etc.

In civil engineering, HS algorithm is used in different optimization problems. Bekdaş and Nigdeli [9] found the optimum parameters of tuned mass dampers

(TMDs) under seismic excitations using HS. Niğdeli and Bekdaş [10] designed optimum reinforced concrete shear walls using HS. Jung et al. [11] developed a hybrid HS for water distribution system design. Cakiroglu et al. [12] used HS when maximizing the buckling load of dispersed laminated composite plates with rectangular geometry. Kayabekir et al. [13] found the minimum total potential energy of the structural system using hybrid HS. Kayabekir et al. [14] optimized the mechanical properties of active tuned mass dampers (ATMDs) and the active controller parameters of proportional-integral-derivative (PID) type controllers using HS. Toklu et al. [15] have optimized the nozzle movement for the additive manufacturing of concrete structures and concrete elements using HS. Haghshenas et al. [16] used HS for slope stability analysis Yücel et al. [17] performed the optimal design of reinforced concrete (RC) structural elements using adaptive HS. Uray and Çarbaş [18] investigated dynamic loads and different soil properties in the optimal design of cantilever retaining walls using a harmony search algorithm. Kayabekir et al. [19] used adaptive HS for the minimum-cost design of reinforced concrete (RC) columns. Bekdaş et al. [20] proposed a modified HS methodology for the optimization of reinforced concrete beams with minimum CO_2 emissions. Ocak et al. [21] optimized seismic base isolation system using adaptive HS. Bekdaş et al. [22] used HS for the optimum design of prestressed concrete cylindrical walls. Aydın et al. [23] used HS for environmentally friendly optimum design of reinforced concrete columns.

The HS algorithm consists of five basic steps. The first step is to set up the problem and set the algorithm parameters. The optimization problem is defined as is shown in Eq. 1. $f(x)$ is the objective function to be minimized, x_i is the decision variables, X_i is the solution space used for the i. decision variable, N is the total number of decision variables.

$$z = min\{f(x)\} x_i \epsilon X_i i = 1, 2, 3, \ldots, N \tag{1}$$

The parameters of the HS algorithm are Harmony Memory Search (HMS), Harmony Memory Considering Rate (HMCR), Pitch Adjusting Rate (PAR). HMS is the number of solutions with the best result to be kept in memory at each iteration while the algorithm runs. HMCR indicates the proportion of the harmony memory to be taken into account when creating a new harmony. PAR indicates the rate at which tone adjustment will be applied to the values selected from memory.

The second step is the creation of harmony memory. A random harmony matrix (HM) of the size of the HMS is generated. In this generated matrix (Eq. 2), each row represents a solution and each column represents a design parameter. The vector f(x) represents the objective function corresponding to each solution.

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \ldots & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \ldots & x_N^2 & f(x^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \ldots & x_N^{HMS} & f(x^{HMS}) \end{bmatrix} \tag{2}$$

The third step is the creation of a new harmony. In this step, a randomly generated value (r) between zero and 1 is calculated according to whether it is greater or less than the HMCR value (Eq. 3):

$$X_{i,new} = \begin{cases} X_{i,min} + rand()\left(X_{i,max} - X_{i,min}\right) \; if \; HMCR > r_1 \\ X_{i,k} + PAR\left(X_{i,max} - X_{i,min}\right) \; if \; HMCR \leq r_1 \end{cases} \tag{3}$$

$$k = ceil(rand * HMS) \tag{4}$$

k in Eq. 3 is a randomly chosen existing solution as seen in Eq. 4. The fourth step of the algorithm is to update the harmony memory. The newly developed harmony is compared with the worst harmony in the HM matrix and the worst harmony is removed from the HM matrix [3].

The fifth step is to check the stopping condition. Step 3 and step 4 are continued in cycles until the termination criterion is met, and the process is terminated when the termination criterion is met. Figure 1 shows the HS flow chart.

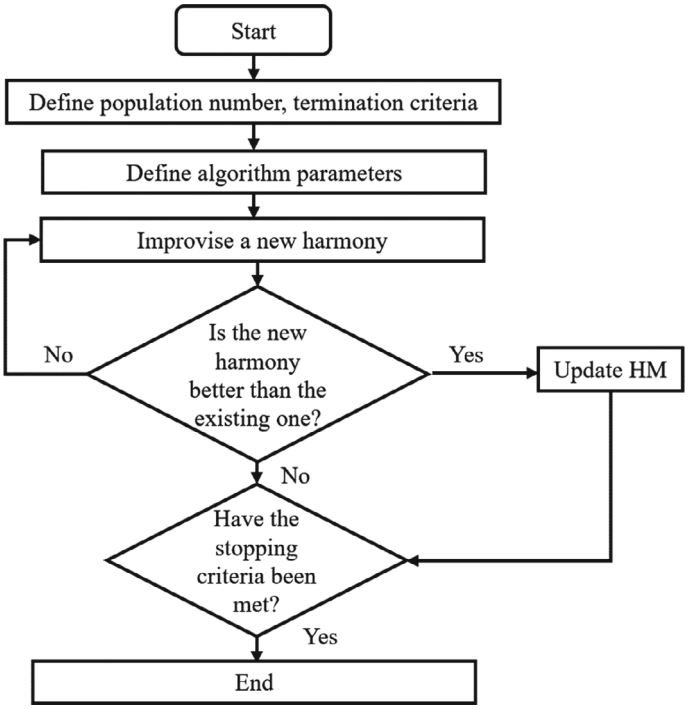


Fig. 1 HS flowchart [14]

2.2 Categorical Boosting (CatBoost)

Machine Learning (ML) is a branch of artificial intelligence that enables computers to think and learn on their own. One of the fields where machine learning is widely used is civil engineering. Machine learning can be used in hydraulics [24–26], geotechnics [27–33], materials [34–37], structural optimization [22, 23, 38], construction planning [23], etc.

In this book chapter, firstly rectangular section reinforced concrete columns under uniaxial bending effect data generated with the HS algorithm and used in the analysis with CatBoost. Thus, the effects of CatBoost parameters on the prediction success of the CatBoost model are investigated.

CatBoost was developed by Yandex researchers and engineers for internal use [22]. The CatBoost model is a machine learning model based on a gradient boosting algorithm using the boosting technique, where each tree is trained by focusing on correcting the errors of the previous trees. This results in an overall more powerful and performant prediction model. Unlike other gradient-boosting algorithms, CatBoost specifically focuses on the direct use of categorical variables. CatBoost can process categorical variables automatically. Catboost is an open-source project and uses many programming languages, including Python. Catboost has the ability to make fast and effective predictions thanks to an optimized prediction algorithm after training. Due to its better handling of categorical variables, the data preparation process is shortened and model performance is improved. It exhibits strong performance, especially in data sets with a high number of categorical variables [39]. In the CatBoost model, several weak learners are created by training a weighted training set and feedback is given to correct the weighting. In the final stage, a strong learner is reached based on multiple learners. A diagram of CatBoost is given in Fig. 2 [40].

The CatBoost algorithm has more than one parameter. The parameters used in this study are given in Table 1. The CatBoost parameters used in this study are explained in the rest of the paper. The "iterations" parameter determines the maximum number of boosting iterations (trees) to be used during training. The parameter "learning_rate" controls the step size in each iteration on the way to the minimum loss function. The "depth" parameter determines the maximum depth of trees in the ensemble. The parameter "l2_leaf_reg" determines the coefficient of the cost function in the

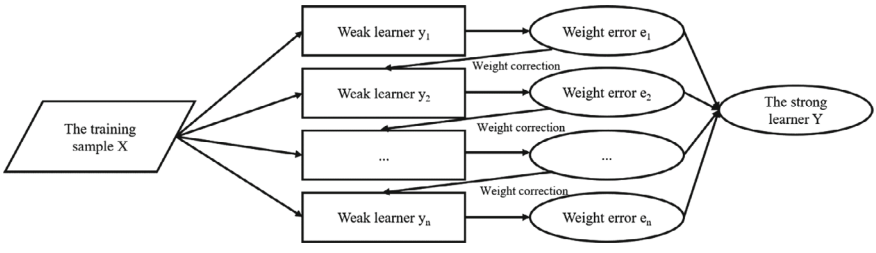


Fig. 2 Diagram of CatBoost structure [40]

Table 1 Parameters used in the study

Parameter	Description
"iterations"	Maximum number of boosting iterations
"learning_rate"	Controls the step size in each iteration
"depth"	Maximum depth of trees
"l2_leaf_reg"	Coefficient of the cost function
"random_seed"	Number of random seeds
"loss_function"	Metric

L2 regularization period. The parameter "random_seed" determines the number of random seeds used for training. The parameter "loss_function" determines the metric to be used in training [41].

2.2.1 Performance Metrics

In this study, the coefficient of determination (R^2) is used for performance evaluation. The coefficient of determination is the variance of the dependent variable explained by the independent variables. A coefficient of determination of zero indicates that the independent variables can't explain the dependent variable at all, while a coefficient of one indicates that they can fully explain it.

$$R^2 = \frac{Variance\ explained\ by\ the\ model}{Total\ variance} \tag{5}$$

Figure 3 explains the proposed method in this study.

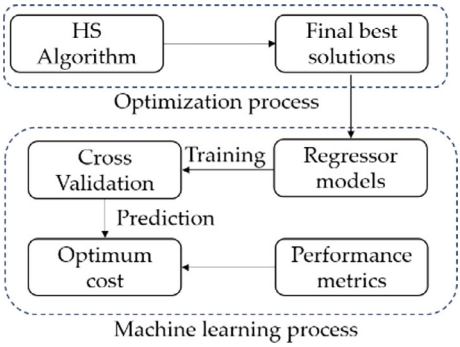


Fig. 3 Flowchart of the proposed method

3 Economic Design of Column

In this chapter, the cost minimization of a rectangular reinforced concrete column under uniaxial bending is carried out and the obtained data is used in machine learning. Figure 4 shows a reinforced concrete column under uniaxial bending with bending moment M_y and axial load N_z.

The aim of this problem is to find the design with the lowest cost that satisfies the safety conditions specified in the theoretical and regulatory codes. The cost of the column is the sum of the cost of concrete and steel used in its design. The objective function that defines this is given in Eq. 5.

$$f(x) = (bh - \sum A_s)lC_c + \left(\sum A_s\right)l\gamma_s C_s \tag{5}$$

In Eq. 5, b is the section width, h is the section height, A_s is the reinforcement cross-sectional area, l is the column length and C_c is the unit cost of concrete m^3. The first part of the equation constitutes the total concrete cost. γs is the unit volume weight of steel and C_s is the unit cost in tons of steel. Since the values of unit cost, length and unit volume weight of material are constant in the equation, the variables that will change the cost are section dimensions and reinforcement cross-sectional area.

In the problem, column length (l) 3 m, axial force (N) 2000 kN, bending moment (M) 50 kNm, steel yield strength (f_y) 420 Mpa, concrete compressive strength (f'c) 25 Mpa, steel modulus of elasticity (Es) 200,000 MPa, steel unit volume weight (γ_s)

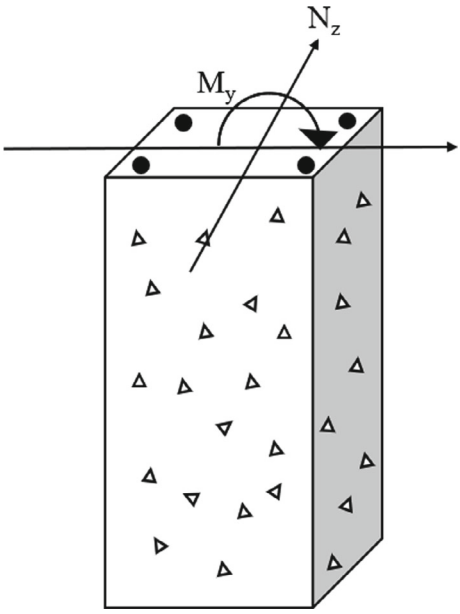


Fig. 4 Forces acting on the wall for static loading condition

Table 2 Design variables

Design variables	Min	Max
Cross-section width (b) (mm)	250	400
Section height (h) (mm)	300	600
Reinforcement ratios (ρ)	0.01	0.06

7. 86 t/m3, concrete unit m3 cost (C_c) 40 units, steel unit ton cost (C_s) 400 units, concrete fracture unit shortening (ε_{cu}) 0.03, reinforcement center to end of section distance (d') 60 mm. The ranges of design variables are given in Table 2.

Figure 5 shows the deformation of a column under axial load and bending moment (M = Ne) and the in-section stresses and forces due to these deformations. Equivalent rectangular stress distribution is taken as a representative of the section under compression.

When the equilibrium equations are written for the forces given in Fig. 5, the axial load and bending moment capacity of the column can be calculated by Eqs. 6 and 7.

$$N_d = 0.85 \times f_c\prime \times a \times b + A_s\prime f_s\prime - A_s f_s \tag{6}$$

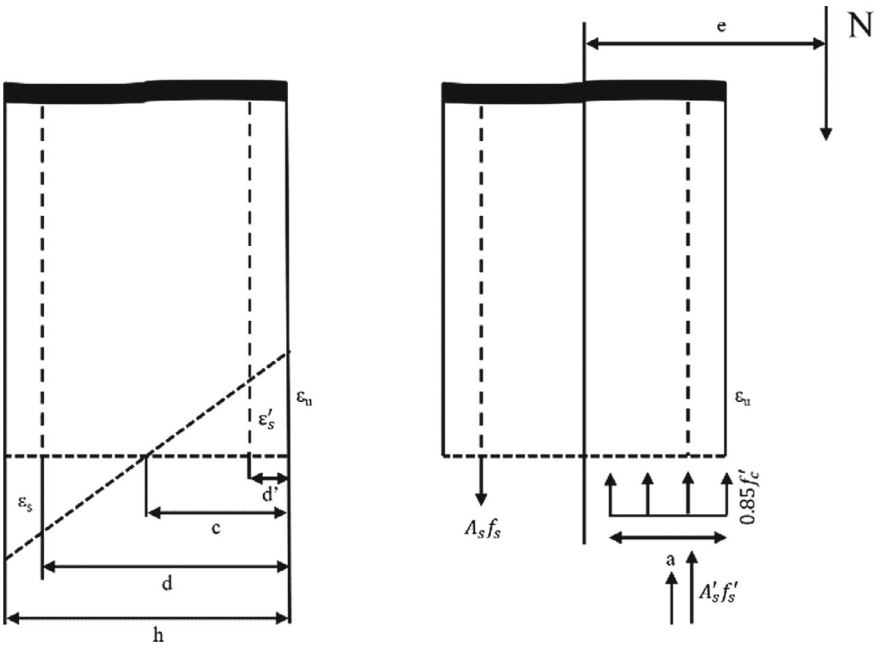


Fig. 5 Deformation, stresses and forces in reinforced concrete column under uniaxial bending effect

Table 3 Design constraints

Description	Constraint
Maximum axial load (N_{max})	$N_d \leq N_{max} = 0.5$
Minimum reinforcement area (As_{min})	$A_s \geq A_{s,min} = 0.01bh$
Maximum reinforcement area (As_{max})	$A_s \leq A_{s,max} = 0.06bh$
Bending moment capacity (M_{dx} ve M_{dy})	$M_{dx} \geq M_{ux}$ and $M_{dy} \geq M_{uy}$
Axial force capacity (N_d)	$N_d \geq N_u$

$$M_d = 0.85 \times f_c\prime \times a \times b \times (\frac{h}{2} - \frac{a}{2}) + A_s\prime \times f_s\prime \times (\frac{h}{2} - d\prime) - A_s \times f_s \times (d - \frac{h}{2}) \tag{7}$$

The f_s and f_s' values in Eqs. 6 and 7 are the stresses calculated based on the ε_s and ε_s' stresses in the reinforcement. The a value can be calculated by Eq. 8 for the equivalent stress block depth based on the neutral axis (c) distance.

$$a = c \times \beta_1 \tag{8}$$

In Eq. 8, β_1 is a coefficient related to the characteristic compressive strength of concrete and this value is defined by Eq. 9 in ACI 318 Building Code Requirements for Reinforced Concrete [42].

$$\beta_1 = \begin{cases} 0.8517 MPa < f_c\prime \leq 28 MPa \\ 0.85 - 0.0071428(f_c\prime - 28) f_c\prime \geq 28 MPa \end{cases} \tag{9}$$

After the reinforced concrete section capacity calculation, the limits in the regulation should be checked to ensure section ductility. These controls are given in Table 3.

4 Results

6 different CatBoostRegressor parameters were used in the study and their effects on the success of the model are given in 3 tables. In Table 4, the effects of "iterations" and "learning_rate" parameters are analysed. When the "iterations" parameter is analysed in Table 4, it is seen that as the value of "iterations" increases, R^2 increases. When the "learning_rate" parameter is analysed, it is seen that different values give slightly the same R^2 values.

In Table 5 the effects of "depth" and "l2_leaf_reg" parameters are analysed. When the "depth" parameter is analysed in Table 5, a value of 2 gives the highest accuracy. When the "l2_leaf_reg" parameter is analysed, it is seen that different values give slightly the same R^2 values.

Table 4 Effects of the iterations and learning_rate parameters

Parameters	Model Architecture	R^2
Default	iterations = “None”, learning_rate = “None”, depth = “ None”, l2_leaf_reg = “ None”, random_seed = “None “ and loss_function = “RMSE”	0.993226765476558
iterations	iterations = “5”, learning_rate = “None”, depth = “ None”, l2_leaf_reg = “None”, random_seed = “ None “ and loss_function = “RMSE”	0.9696831950973414
	iterations = “10”, learning_rate = “None”, depth = “ None”, l2_leaf_reg = “None”, random_seed = “ None “ and loss_function = “RMSE”	0.9953046054859945
	iterations = “50”, learning_rate = “None”, depth = “ None”, l2_leaf_reg = “None”, random_seed = “ None “ and loss_function = “RMSE”	0.9954906007552426
learning_rate	iterations = “None”, learning_rate = “0.3”, depth = “ None”, l2_leaf_reg = “None”, random_seed = “ None “ and loss_function = “RMSE”	0.9937848526268551
	iterations = “None”, learning_rate = “0.8”, depth = “ None”, l2_leaf_reg = “None”, random_seed = “ None “ and loss_function = “RMSE”	0.9937848526134476
	iterations = “None”, learning_rate = “1”, depth = “1”, l2_leaf_reg = “1”, random_seed = “ None “ and loss_function = “RMSE”	0.993784852643908

Table 5 Effects of the depth and l2_leaf_reg parameters

Parameters	Model Architecture	R^2
Default	iterations = "None", learning_rate = "None", depth = " None", l2_leaf_reg = " None", random_seed = " None " and loss_function = "RMSE""	0.993226765476558
depth	iterations = "None", learning_rate = "None", depth = "2", l2_leaf_reg = " None", random_seed = " None" and loss_function = "RMSE"	0.9962724212737869
	iterations = "None", learning_rate = "None", depth = "5", l2_leaf_reg = " None", random_seed = " None" and loss_function = "RMSE"	0.9933693656029737
	iterations = "None", learning_rate = "None", depth = "10", l2_leaf_reg = " None", random_seed = " None" and loss_function = "RMSE"	0.9932800810234905
l2_leaf_reg	iterations = "None", learning_rate = "0.3", depth = "None", l2_leaf_reg = "1", random_seed = " None " and loss_function = "RMSE"	0.993764896986681
	iterations = " None", learning_rate = "0.8", depth = "None", l2_leaf_reg = "4", random_seed = " None " and loss_function = "RMSE"	0.9933080604863002
	iterations = " None", learning_rate = "1", depth = "None", l2_leaf_reg = "10", random_seed = " None " and loss_function = "RMSE"	0.9979028489328047

In Table 6 the effects of "random_seed" and "loss_function" parameters are analysed. When the "random_seed" parameter is analysed in Table 6, it is seen that different values give slightly the same R^2 values. When the "loss_function" parameter is analysed from Table 6, MAPE gives the highest R^2.

5 Conclusion

In this study, firstly, the cost optimization of a rectangular reinforced concrete column under uniaxial bending is performed with the optimisation process. The Categorical Boosting (CatBoost) Regressor model is used to predict the cost of the rectangular reinforced concrete column under uniaxial bending on the 1000-row dataset obtained as a result of the optimization and the effect of CatBoost parameters on performance is examined. Coefficient of determination (R^2) was used in performance evaluation.

The parameters whose effects on the model are examined in the study are "iterations", "learning_rate", "depth", "l2_leaf_reg", "random_seed" and "loss_function".

Among the parameters used in the study, the "iterations" parameter had the highest effect. A value of 5 of this parameter decreased the R^2 value of the default model. 5 iterations means that the CatBoost algorithm will run 5 times to minimize the loss function. Therefore, a lower value indicates that the algorithm will work less.

This study demonstrates the success of parameter variation on the model through the example of minimum cost of rectangular reinforced concrete columns under uniaxial bending.

Table 6 Effects of the random_seed and loss_function parameters

Parameters	Model Architecture	R^2
Default	iterations = " None ", learning_rate = "None", depth = " None", l2_leaf_reg = "None", random_seed = " None " and loss_function = "RMSE"	0.993226765476558
random_seed	iterations = "None ", learning_rate = "None", depth = " None", l2_leaf_reg = " None", random_seed = "1" and loss_function = "RMSE"	0.9931648687673296
	iterations = "None ", learning_rate = "None", depth = " None", l2_leaf_reg = " None", random_seed = "20" and loss_function = "RMSE"	0.9932570010990637
	iterations = "None ", learning_rate = "None", depth = "40", l2_leaf_reg = " None", random_seed = " 50 " and loss_function = "RMSE"	0.9931186561570449
loss_function	iterations = "None ", learning_rate = "None", depth = " None", l2_leaf_reg = " None", random_seed = "None " and loss_function = "MAE""	0.9936234279974976
	iterations = "None ", learning_rate = "None", depth = " None", l2_leaf_reg = "None", random_seed = "None " and loss_function = "MAPE"	0.9951046786624019

References

1. Celep, Z., Kumbasar, N. (2001). Reinforced concrete structures. İhlas Printing.
2. Nigdeli, S.M., Bekdaş, G., Kayabekir, A.E.: Optimization of Reinforced Concrete Building Elements with Meta-Intuitive Methods. IUC Press, Istanbul (2023). https://doi.org/10.5152/2904
3. Geem, Z.W., Kim, J.H., Loganathan, G.V.: New heuristic optimization algorithm: Harmony search. Simulation **76**(2), 60–68 (2001). https://doi.org/10.1177/003754970107600201
4. Al-Betar, M.A., Khader, A.T.: A harmony search algorithm for university course timetabling. Ann. Oper. Res. **194**, 3–31 (2012). https://doi.org/10.1007/s10479-010-0769-z
5. Diao, R., Shen, Q. (2012). Feature selection with harmony search. IEEE Trans. Syst., Man, Cybern., Part B (Cybern.) 42(6), 1509–1523. https://doi.org/10.1109/TSMCB.2012.2193613
6. Güven, A.F., Yörükeren, N., Samy, M.M.: Design optimization of a stand-alone green energy system of university campus based on Jaya-Harmony search and ant colony optimization algorithms approaches. Energy **253**, 124089 (2022). https://doi.org/10.1016/j.energy.2022.124089
7. Huang, Y.F., Chen, P.H.: Fake news detection using an ensemble learning model based on self-adaptive harmony search algorithms. Expert. Syst. Appl. **159**, 113584 (2020). https://doi.org/10.1016/j.eswa.2020.113584
8. Attuluri, S., Ramesh, M.: Multi-objective discrete harmony search algorithm for privacy preservation in cloud data centers. Int. J. Inf. Technol. **15**(8), 3983–3997 (2023). https://doi.org/10.1007/s41870-023-01462-w
9. Bekdaş, G., Nigdeli, S.M.: Estimating optimum parameters of tuned mass dampers using harmony search. Eng. Struct. **33**(9), 2716–2723 (2011). https://doi.org/10.1016/j.engstruct.2011.05.024
10. Nigdeli, S.M., Bekdaş, G. (2014). Optimization of reinforced concrete shear walls using harmony search. In: 11th International Congress on Advances in Civil Engineering (pp. 21–25).
11. Jung, D., Kang, D., Kim, J.H.: Development of a hybrid harmony search for water distribution system design. KSCE J. Civ. Eng. **22**, 1506–1514 (2018). https://doi.org/10.1007/s12205-017-1864-3
12. Cakiroglu, C., Bekdaş, G., Geem, Z.W. (2020). Harmony search optimisation of dispersed laminated composite plates. Materials 13(12), 2862. c10.3390/ma13122862
13. Kayabekir, A.E., Toklu, Y.C., Bekdaş, G., Nigdeli, S.M., Yücel, M., Geem, Z.W.: A novel hybrid harmony search approach for the analysis of plane stress systems via total potential optimization. Appl. Sci. **10**(7), 2301 (2020). https://doi.org/10.3390/app10072301
14. Kayabekir, A.E., Bekdaş, G., Nigdeli, S.M., Geem, Z.W.: Optimum design of PID controlled active tuned mass damper via modified harmony search. Appl. Sci. **10**(8), 2976 (2020). https://doi.org/10.3390/app10082976
15. Toklu, Y.C., Bekdaş, G., Geem, Z.W.: Harmony search optimization of nozzle movement for additive manufacturing of concrete structures and concrete elements. Appl. Sci. **10**(12), 4413 (2020). https://doi.org/10.3390/app10124413
16. Haghshenas, S.S., Haghshenas, S.S., Geem, Z.W., Kim, T.H., Mikaeil, R., Pugliese, L., Troncone, A.: Application of harmony search algorithm to slope stability analysis. Land **10**(11), 1250 (2021). https://doi.org/10.3390/land10111250
17. Yücel, M., Kayabekir, A.E., Bekdaş, G., Nigdeli, S.M., Kim, S., Geem, Z.W.: Adaptive-hybrid harmony search algorithm for multi-constrained optimum eco-design of reinforced concrete retaining walls. Sustainability **13**(4), 1639 (2021). https://doi.org/10.3390/su13041639
18. Uray, E., Çarbaş, S. (2021). Dynamic loads and different soil characteristics examination on optimum design of cantilever retaining walls utilizing harmony search algorithm. Int. J. Eng. Appl. Sci. 13(4), 140–154. https://doi.org/10.24107/ijeas.1033802
19. Kayabekir, A.E., Nigdeli, S.M., Bekdaş, G. (2022). Adaptive harmony search for cost optimization of reinforced concrete columns. In: Intelligent Computing & Optimization: Proceedings of the 4th International Conference on Intelligent Computing and Optimization 2021 (ICO2021)

3 (pp. 35–44). Springer International Publishing. https://doi.org/10.1007/978-3-030-93247-3_4
20. Bekdaş, G., Nigdeli, S.M., Kim, S., Geem, Z.W.: Modified harmony search algorithm-based optimization for eco-friendly reinforced concrete frames. Sustainability **14**(6), 3361 (2022). https://doi.org/10.3390/su14063361
21. Ocak, A., Nigdeli, S.M., Bekdaş, G., Kim, S., Geem, Z.W.: Optimization of seismic base isolation system using adaptive harmony search algorithm. Sustainability **14**(12), 7456 (2022). https://doi.org/10.3390/su14127456
22. Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z.W.: Optimal dimensions of post-tensioned concrete cylindrical walls using harmony search and ensemble learning with SHAP. Sustainability **15**(10), 7890 (2023). https://doi.org/10.3390/su15107890
23. Aydın, Y., Bekdaş, G., Nigdeli, S.M., Isıkdağ, Ü., Kim, S., Geem, Z.W.: Machine learning models for ecofriendly optimum design of reinforced concrete columns. Appl. Sci. **13**(7), 4117 (2023). https://doi.org/10.3390/app13074117
24. Hosseiny, H., Nazari, F., Smith, V., Nataraj, C.: A framework for modeling flood depth using a hybrid of hydraulics and machine learning. Sci. Rep. **10**(1), 8222 (2020). https://doi.org/10.1038/s41598-020-65232-5
25. Kalumba, M., Nyirenda, E., Nyambe, I., Dondeyne, S., Van Orshoven, J.: Machine learning techniques for estimating hydraulic properties of the topsoil across the Zambezi River Basin. Land **11**(4), 591 (2022). https://doi.org/10.3390/land11040591
26. Astsauri, T., Habiburrahman, M., Ibrahim, A.F., Wang, Y.: Utilizing machine learning for flow zone indicators prediction and hydraulic flow unit classification. Sci. Rep. **14**(1), 4223 (2024). https://doi.org/10.1038/s41598-024-54893-1
27. Puri, N., Prasad, H.D., Jain, A.: Prediction of geotechnical parameters using machine learning techniques. Procedia Comput. Sci. **125**, 509–517 (2018). https://doi.org/10.1016/j.procs.2017.12.066
28. Rauter, S., Tschuchnigg, F.: CPT data interpretation employing different machine learning techniques. Geosciences **11**(7), 265 (2021). https://doi.org/10.3390/geosciences11070265
29. Nanehkaran, Y.A., Licai, Z., Chengyong, J., Chen, J., Anwar, S., Azarafza, M., Derakhshani, R.: Comparative analysis for slope stability by using machine learning methods. Appl. Sci. **13**(3), 1555 (2023). https://doi.org/10.3390/app13031555
30. Aydın, Y., Işıkdağ, Ü., Bekdaş, G., Nigdeli, S.M., Geem, Z.W.: Use of machine learning techniques in soil classification. Sustainability **15**(3), 2374 (2023). https://doi.org/10.3390/su15032374
31. Aydın, Y., Bekdaş, G., Isikdag, U., Nigdeli, S.M., Geem, Z.W. (2024). Optimizing Artificial Neural Network Architectures for Enhanced Soil Type Classification, Geomechanics and Engineering. An Int. J. 37(3), 263–277. https://doi.org/10.12989/gae.2024.37.3.263
32. Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z.W.: Optimal dimensioning of retaining walls using explainable ensemble learning algorithms. Materials **15**(14), 4993 (2022). https://doi.org/10.3390/ma15144993
33. Cakiroglu, C., Islam, K., Bekdaş, G., & Nehdi, M.L. (2023). Data-driven ensemble learning approach for optimal design of cantilever soldier pile retaining walls. In: Structures (Vol. 51, pp. 1268–1280). Elsevier. https://doi.org/10.1016/j.istruc.2023.03.109
34. Chan, C.H., Sun, M., Huang, B.: Application of machine learning for advanced material prediction and design. EcoMat **4**(4), e12194 (2022). https://doi.org/10.1002/eom2.12194
35. Aydın, Y., Cakiroglu, C., Bekdaş, G., Işıkdağ, Ü., Kim, S., Hong, J., Geem, Z.W.: Neural network predictive models for alkali-activated concrete carbon emission using metaheuristic optimization algorithms. Sustainability **16**(1), 142 (2023). https://doi.org/10.3390/su16010142
36. Cakiroglu, C., Aydın, Y., Bekdaş, G., Geem, Z.W.: Interpretable predictive modelling of basalt fiber reinforced concrete splitting tensile strength using ensemble machine learning methods and SHAP approach. Materials **16**(13), 4578 (2023). https://doi.org/10.3390/ma16134578
37. Cakiroglu, C., Bekdaş, G.: Predictive Modeling of recycled aggregate concrete beam shear strength using explainable ensemble learning methods. Sustainability **15**(6), 4957 (2023). https://doi.org/10.3390/su15064957

38. Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z.W.: Optimization and predictive modeling of reinforced concrete circular columns. Materials **15**(19), 6624 (2022). https://doi.org/10.3390/ma15196624
39. Feng, K., Chen, S., Lu, W. (2018). Machine learning based construction simulation and optimization. In: 2018 Winter Simulation Conference (WSC) (pp. 2025–2036). IEEE. https://doi.org/10.1109/WSC.2018.8632290
40. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Adv. Neural Inf. Process. Syst. 31
41. Training parameters, https://catboost.ai/en/docs/references/training-parameters/
42. ACI-318.: Building code requirements for structural concrete and commentary, metric version. Am. Concr. Inst. (2005)

Machine Learning Approaches for Predicting Compressive and Shear Strength of EB FRP-Reinforced Concrete Elements: A Comprehensive Review

Ali Benzaamia, Mohamed Ghrici, and Redouane Rebouh

Abstract The reinforcement of concrete structures with externally bonded fiber-reinforced polymer (EB FRP) composites has emerged as an innovative technique for enhancing structural performance, durability, and service life, yet accurately predicting the shear strength of these elements remains complex due to intricate interactions between influencing factors that conventional empirical equations struggle to capture. This comprehensive review critically analyzes state-of-the-art machine learning (ML) approaches as data-driven alternatives to model the strength behavior of EB FRP-reinforced concrete elements with improved accuracy and reliability. It encompasses techniques including artificial neural networks (ANNs), support vector machines (SVMs), genetic programming (GP), ensemble learning (random forests, gradient boosting), and emerging methods like gene expression programming (GEP) and multigene genetic programming (MGP). By synthesizing the latest advances, evaluating different ML approaches, and identifying gaps and future directions, this review aims to provide a comprehensive understanding of ML's potential and limitations in predicting the compressive and shear strength of EB FRP-reinforced concrete elements.

Keywords Concrete · FRP · Compressive strength · Shear strength · Machine learning · Artificial neutral network

Please note that the AISC Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

A. Benzaamia · M. Ghrici (✉) · R. Rebouh
Geomaterials Laboratory, University Hassiba Benbouali of Chef, Ouled Fares, Algeria
e-mail: m.ghrici@univ-chlef.dz

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_12

1 Introduction

The reinforcement of concrete structures with externally bonded fiber-reinforced polymer (EB FRP) composites has emerged as an innovative and effective technique for enhancing their structural performance, durability, and service life [1, 2]. By leveraging the superior tensile strength, lightweight nature, and corrosion resistance of FRP materials, this strengthening approach has gained widespread adoption in the rehabilitation and retrofitting of existing concrete structures, as well as in the design of new constructions [3–5]. The application of EB FRP systems offers numerous advantages, including increased load-bearing capacity, improved ductility and deformation characteristics, and enhanced resistance to environmental degradation mechanisms [6].

However, accurately predicting the compressive and shear strength of EB FRP-reinforced concrete elements remains a complex challenge due to the intricate interactions between the various influencing factors. These factors encompass concrete properties, FRP reinforcement characteristics, interface bond behavior, geometric configurations, and loading conditions. The complexity arises from the nonlinear behavior exhibited by these structural systems, involving phenomena such as concrete confinement, shear transfer mechanisms, and potential failure modes (e.g., FRP rupture, debonding) [7–10].

Conventional design approaches have primarily relied on empirical equations derived from experimental data and simplified analytical models based on assumed failure mechanisms. While these methods have facilitated the widespread implementation of design codes and guidelines, their accuracy and reliability are often limited by the inherent assumptions and approximations employed [11, 12]. The complex failure mechanisms and highly nonlinear behavior exhibited by EB FRP-reinforced concrete elements pose significant challenges for these traditional approaches, potentially leading to overly conservative or unconservative designs, which can result in either uneconomical solutions or compromised structural safety.

In recent decades, the rapid advancement of machine learning (ML) techniques has opened new avenues for addressing complex engineering problems by leveraging data-driven modeling and computational intelligence. ML algorithms, inspired by the human brain's ability to learn from data, have demonstrated remarkable capabilities in capturing intricate patterns and nonlinear relationships from large datasets, making them well-suited for predicting the behavior of intricate structural systems [13–16]. By employing ML techniques to model the compressive and shear strength of EB FRP-reinforced concrete elements, researchers and practitioners can potentially overcome the limitations of conventional empirical models and achieve more accurate and reliable strength predictions.

This comprehensive review aims to provide a critical analysis of the state-of-the-art application of ML approaches to predict the compressive and shear strength of EB FRP-reinforced concrete elements. It encompasses a wide range of ML techniques, including artificial neural networks (ANNs), support vector machines (SVMs), genetic programming (GP), ensemble learning methods (e.g.,

random forests, gradient boosting), and emerging techniques such as gene expression programming (GEP) and multigene genetic programming (MGP). The review evaluates the strengths, limitations, and predictive capabilities of these ML models, highlighting their performance in comparison to conventional empirical models and design guidelines.

Furthermore, the review examines the influence of various input parameters on the accuracy and robustness of the ML models. These input parameters include concrete properties (compressive strength, elastic modulus), FRP reinforcement characteristics (type, thickness, elastic modulus, ultimate strain), geometric configurations (cross-sectional shape, dimensions, aspect ratios), and loading conditions. The review also discusses the challenges associated with data availability, model optimization, and reliability assessment of ML techniques in the context of strength prediction for EB FRP-reinforced concrete structures.

By synthesizing the latest research advances, evaluating the performance of different ML approaches, and identifying current gaps and future research directions, this review aims to provide a comprehensive understanding of the potential and limitations of ML techniques in predicting the compressive and shear strength of EB FRP-reinforced concrete elements. Ultimately, this knowledge can contribute to the development of more accurate and reliable strength prediction models, facilitating the efficient design, assessment, and rehabilitation of these structural systems while ensuring safety, durability, and cost-effectiveness.

2 Machine Learning: Concepts and Methodologies

2.1 Machine Learning, AI, and Deep Learning

The concepts of Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are often misunderstood within the civil engineering community. AI is a broad field encompassing algorithms and techniques that aim to mimic human intelligence for tasks like reasoning, decision-making, and problem-solving [17–19]. ML is a subfield of AI involving algorithms that enable machines to learn from data to make predictions and classifications. DL is a specialized branch of ML that utilizes sophisticated neural network architectures capable of learning from large datasets and tackling complex problems. Figure 1 clearly illustrates the distinctions between AI, ML, and DL.

2.2 Major Categories of Machine Learning Algorithms

Machine learning (ML) is a subfield of artificial intelligence that develops models capable of learning from data to make forecasts or classifications without explicit

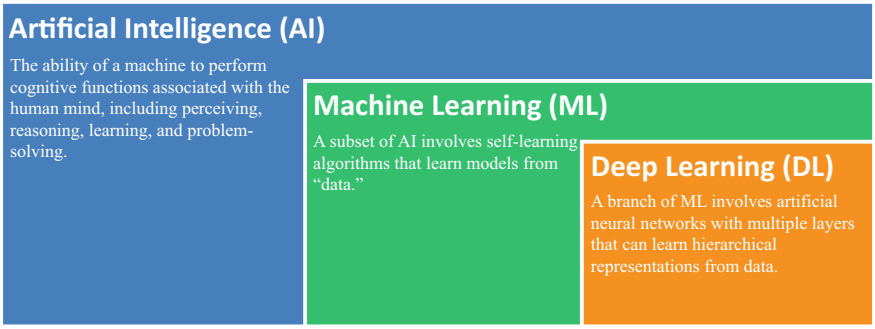


Fig. 1 Artificial intelligence, machine learning, and deep learning

programming [20]. The primary goal of ML is to establish mathematical models, also known as "black boxes," trained on datasets to make predictions or decisions on new data [21]. ML models have been widely applied in various domains, including image analysis, data regression, experimental modeling, natural language processing, and medical diagnostics [22]. ML can be categorized into four main types, as illustrated in Fig. 2:

- **Supervised Learning**: This category relies on labeled training instances derived from numerical or experimental data [23, 24]. The aim is to determine the correlations between independent and dependent variables, enabling predictions or classifications of the dependent variable. Supervised learning paradigms include linear regression, artificial neural networks, support vector machines, and decision trees.
- **Unsupervised Learning**: These algorithms utilize unlabeled data instances and are primarily employed for data clustering, anomaly detection, and dimension

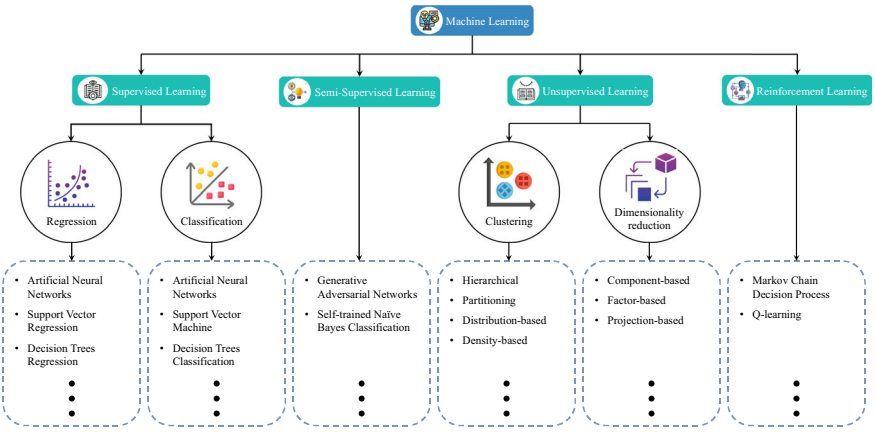


Fig. 2 Categories of Machine Learning Algorithms

reduction. Clustering divides data into groups based on similarity, while anomaly detection identifies abnormal data points. Dimension reduction eliminates unnecessary features without affecting input–output correlations [25].

- **Semi-Supervised Learning**: This approach combines labeled and unlabeled training instances. It aims to leverage the labeled data to enhance the model's performance in learning patterns from unlabeled instances. Semi-supervised learning addresses the limitations of supervised and unsupervised learning by utilizing a small portion of labeled data and a large part of unlabeled data, effectively decreasing the time needed for data preprocessing [26].
- **Reinforcement Learning**: This category relies on agents operating in an environment, learning through a trial-and-error process using feedback from their experiences [27]. Unlike supervised learning, reinforcement learning employs a reward and punishment system for desirable and undesirable actions. While widely used in AI applications, reinforcement learning has limited application in structural engineering or concrete science [28, 29].

2.3 *Key Steps in Constructing Effective Machine Learning Models*

The implementation of ML models typically involves four key steps: data collection, data preprocessing, model training, and model testing and evaluation [30], as outlined in Fig. 3:

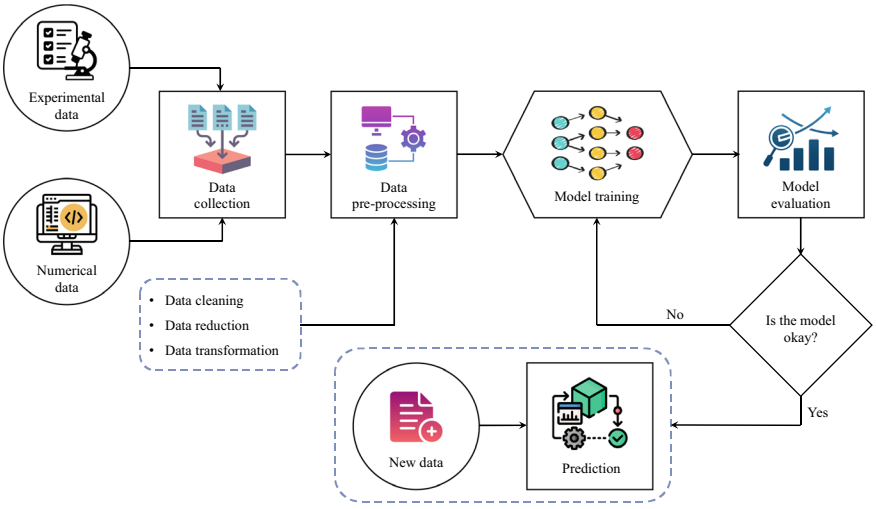


Fig. 3 ML models overall workflow

- **Data Collection**: Collecting a suitable dataset is crucial for developing effective ML models. The dataset should provide necessary learning patterns while considering relevant parameter settings and leveraging the largest available data instances.
- **Data Preprocessing**: This critical step enhances data quality and reliability for ML training. It involves data cleansing (handling missing and noisy data), transformation (normalization, feature selection, discretization), and reduction techniques (dimensionality reduction, sampling) [31, 32].
- **Model Development**: After preprocessing, the data is ready for ML training. The appropriate algorithm (e.g., neural networks, support vector machines, decision trees) is selected based on data characteristics. Training involves partitioning data into training and validation sets, using techniques like holdout or K-fold cross-validation [33]. Performance metrics guide model tuning and refinement.
- **Model Evaluation and Testing**: The trained model's performance is thoroughly evaluated using statistical metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-squared (R2), and Mean Absolute Percentage Error (MAPE). If the performance does not meet the desired accuracy threshold, the model is tuned by expanding the dataset, improving the architecture, or adjusting hyperparameters [34, 35].

Developing high-performing ML models requires careful data selection, robust preprocessing, appropriate algorithm selection, comprehensive performance assessment, and iterative model refinement to achieve accurate predictions and practical real-world applications.

3 Overview of ML Algorithms

This section provides an overview of the various machine learning algorithms that have been employed to predict the compressive and shear strength of EB FRP-reinforced concrete elements. These include regression techniques, artificial neural networks, genetic programming, adaptive neuro-fuzzy inference systems, support vector machines, decision trees, ensemble methods like random forests, and boosting algorithms. The working principles, strengths, and limitations of these algorithms are discussed in detail.

3.1 Regression Techniques

Regression analysis models the relationship between independent variable(s) (predictors) and a dependent variable (target). For predicting the compressive and shear strength of FRP-reinforced concrete elements, regression techniques relate input

parameters like concrete properties, FRP reinforcement details, and geometric configurations to the target strength characteristics.

- **Regression**: Assumes a linear relationship between inputs and target, fitting a straight line/hyperplane to minimize error metrics like MSE or MAE. Includes simple linear (one input) and multiple linear regression.
- **Multivariate Regression**: Extension with multiple dependent variables, helpful for modeling multiple strength characteristics simultaneously.
- **Polynomial Regression**: Extends linear regression by fitting curved polynomials to capture nonlinear relationships but risks overfitting with high-degree polynomials.
- **Regularized Regression**: Techniques like Ridge and Lasso Regression introduce penalties to shrink less important predictor coefficients toward zero, improving generalization and reducing overfitting [36]. Ridge uses L2 regularization, while Lasso uses L1 regularization for automatic variable selection.

While applied in structural engineering, these techniques may have limited performance for highly complex, nonlinear relationships in predicting FRP-reinforced concrete strength. Advanced algorithms like neural networks, support vector machines, and ensemble methods can better capture intricate patterns and nonlinearities.

3.2 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) are powerful machine learning algorithms inspired by biological neural networks, capable of learning complex nonlinear mappings between inputs and outputs [26, 37, 38]. ANNs comprise interconnected nodes (neurons) organized into layers, with weighted connections iteratively adjusted during training to optimize performance. The basic ANN structure consists of an input layer, a hidden layer(s), and an output layer (Fig. 4). Inputs propagate through the network, with each hidden neuron computing a weighted sum transformed by a nonlinear activation function, enabling the modeling of intricate nonlinearities. Optimization algorithms, such as backpropagation (gradient-based optimization), adjust connection weights to minimize prediction errors during training.

While gradient-based optimization dominates ANN training, evolutionary computation (EC) algorithms inspired by natural phenomena have emerged, enhancing training capabilities through global search, escaping local optima, and improving generalization [30]. Hybrid EC-ANN models synergistically combine EC algorithms' search abilities with ANNs' function approximation for improved structural engineering predictions.

ANNs offer several advantages, including their ability to learn from data without relying on explicit programming rules, their capacity to model complex, nonlinear relationships, and their potential for parallel processing, which can lead to computational efficiency. However, they also have limitations, such as the risk of overfitting

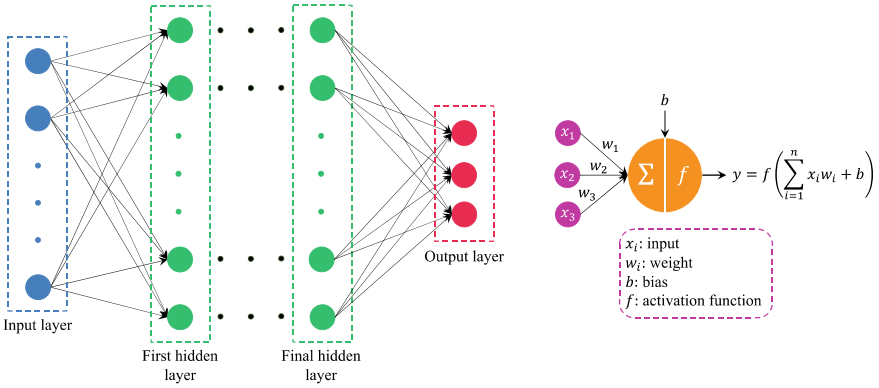


Fig. 4 Simple ANN architecture and its primary component, the neuron

when the model becomes excessively complex and memorizes the training data, the need for large amounts of representative training data, and the potential for interpretability issues, as ANNs can sometimes be perceived as "black box" models.

3.3 Genetic Programming (GP) and Gene Expression Programming (GEP)

Genetic Programming (GP) [39] and Gene Expression Programming (GEP) [40] are two evolutionary computation (EC) techniques that have gained significant attention in the field of structural engineering. These methods draw inspiration from genetic algorithms and utilize evolutionary principles to discover underlying mathematical relationships and patterns within data.

GP employs tree-based representations to evolve computer programs or mathematical expressions that model the target system. The initial population consists of individuals represented as tree structures, with internal nodes as functions and leaf nodes as terminals. Through selection, genetic operations (mutation, crossover), and fitness evaluation, GP evolves these trees to find best-fitting expressions describing input–output relationships. GEP, on the other hand, is an extension of GP that uses a linear representation scheme translatable into expression trees, combining linear and tree-based structure advantages. GEP maintains consistent, compact representations regardless of solution complexity, mitigating GP's bloating problem where trees grow excessively large. GEP has demonstrated superior performance in engineering applications like concrete science due to evolving concise, interpretable mathematical models.

Both GP and GEP are powerful techniques for modeling complex, nonlinear systems, automatically discovering mathematical expressions from data without

explicit programming or prior knowledge. However, performance depends on factors like genetic operator choice, population size, and problem representation.

3.4 Adaptive Neuro-Fuzzy Inference System (ANFIS)

The Adaptive Neuro-Fuzzy Inference System (ANFIS) [41] is a hybrid ML technique combining artificial neural networks and fuzzy inference systems. It leverages neural networks' learning capabilities and fuzzy logic's interpretability to model complex nonlinear relationships. ANFIS is well-suited for imprecise or uncertain data, incorporating fuzzy reasoning. It typically employs the Takagi–Sugeno-Kang (TSK) fuzzy model, where consequent parts of fuzzy rules are linear combinations of input variables, providing a systematic approach to generating rules from data.

The ANFIS architecture consists of five layers: Membership Layer (calculating input membership values), Fuzzification Layer (combining fuzzified inputs), Fuzzy Rule Layer (computing rule firing strengths), Normalization Layer (normalizing firing strengths), and Defuzzification Layer (computing final weighted output). Training involves adjusting membership function parameters and consequent parts to minimize prediction errors through a hybrid learning algorithm combining least-squares and backpropagation methods. ANFIS advantages include capturing nonlinearities while providing interpretable fuzzy rules, aiding pattern understanding. It is computationally efficient and achieves high accuracy in prediction and classification tasks.

3.5 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are powerful ML algorithms for binary classification, regression, and clustering [42–44]. SVMs find an optimal hyperplane that maximizes the margin between classes in a high-dimensional feature space, using support vectors to define the decision boundary. For non-linearly separable data, SVMs employ kernel functions to map inputs into higher dimensions where linear separation is possible. Kernels like linear, polynomial, radial basis function (RBF), and sigmoid enable modeling different nonlinearities [45].

Support Vector Regression (SVR) extends SVMs for regression problems, finding a function approximating targets within a predefined error tolerance while minimizing model complexity and empirical risk.

SVMs effectively handle high-dimensional and sparse data, making them versatile for applications like pattern recognition and signal processing. However, performance depends on kernel choice and tuning hyperparameters like penalty (C) and kernel parameters (e.g., gamma for RBF) through techniques like grid search and cross-validation. SVMs' ability to capture nonlinearities, handle high dimensions,

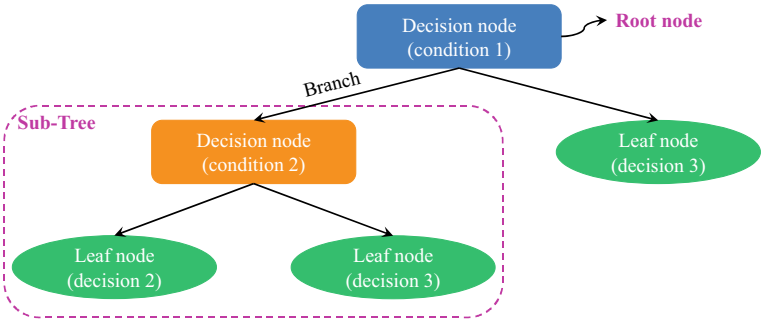


Fig. 5 Example of a DT

and provide sparse solutions makes them powerful tools, but appropriate parameter tuning is crucial for optimal performance.

3.6 Decision Tree (DT)

Decision Trees (DTs) are interpretable ML models representing a series of decisions based on input features to predict outcomes at leaf nodes. They handle numerical and categorical data and offer interpretability and computational efficiency. A DT's structure (Fig. 5) comprises a root node (full dataset), internal decision nodes (splitting data based on features/rules), branches (outcomes from splits), and leaf nodes (final predictions). Tree construction recursively partitions data into subsets at each node using the most discriminative feature based on impurity measures like the Gini index, information gain (classification), or mean squared error (regression) [36].

DTs' key advantage is interpretability, which visualizes the decision process. However, they are prone to overfitting, mitigated by regularization techniques like limiting depth, minimum samples for splitting/leaves, or pruning. Individual DTs may have lower accuracy than ensemble methods like Random Forests or Gradient Boosting, often serving as base learners. Performance depends on impurity measures, stopping criteria, and regularization parameters, requiring proper tuning via techniques like cross-validation or grid search. Despite the simplicity, DTs offer valuable interpretability, while ensemble methods enhance their predictive accuracy.

3.7 Random Forest (RF)

Random Forest (RF) is a powerful ensemble learning technique combining multiple decision trees to improve predictive performance and robustness [46]. RF constructs

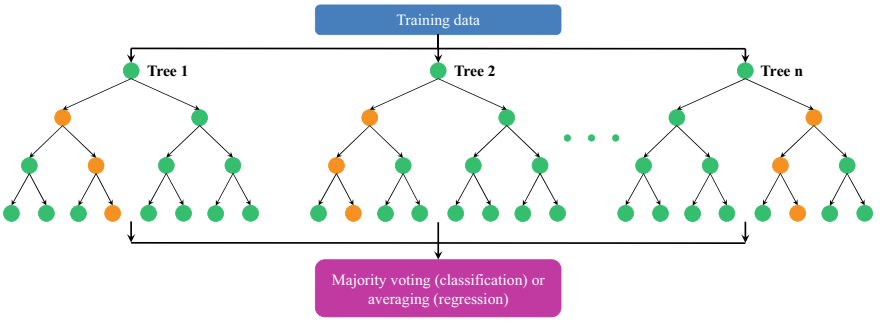


Fig. 6 Parallel RF training: flowchart visualization

an ensemble of trees trained on different subsets of data through bootstrap aggregating (bagging) and random feature selection at each node. The ensemble nature offers advantages over individual trees by reducing variance and overfitting through averaging predictions. Random feature selection decorrelates trees, enhancing diversity and robustness. For prediction, individual tree outputs are combined through majority voting (classification) or averaging (regression) (Fig. 6).

Key strengths include handling high-dimensional data by implicit feature selection during tree growth and relative insensitivity to hyperparameter tuning. However, tuning tree number, depth, and features per split can further improve accuracy. While effective, RF can be computationally intensive during training and has reduced interpretability due to the ensemble nature. Overall, RF leverages ensemble diversity and implicit feature selection to achieve robust, high-dimensional predictive performance, making it a powerful tool for various applications.

3.8 Boosting Algorithm (BA)

Boosting Algorithms are ensemble ML that combine multiple weak learners (e.g., decision trees) to create a strong predictive model [47]. They iteratively train a sequence of weak models, focusing on instances misclassified/poorly predicted by previous models. This additive combination captures complex patterns and nonlinearities for improved predictive performance. Key boosting algorithms include:

- **AdaBoost** (AdaB) [48, 49]: Iteratively trains weak learners (decision stumps) on reweighted data, increasing the weights of misclassified instances. The final model combines weighted weak learners based on performance.
- **Gradient Boosting Machine** (GBM) [50]: Employs gradient descent with decision tree weak learners. At each iteration, a new tree fits the residual errors of the previous ensemble. Offers tuning of trees, learning rate, etc.

- **XGBoost** (XGB) [51]: Efficient, scalable GBM implementation with parallelization, pruning, and compressed data structure. Employs randomization and column subsampling to reduce overfitting.
- **LightGBM** (LGBM) [52]: Efficient GBM focused on computational speed and scalability. Uses leaf-wise tree growth, gradient-based sampling, and feature bundling.
- **CatBoost** (CatB) [53]: Designed for efficient categorical data handling through minimal variance sampling at the tree level.

Boosting algorithms have demonstrated remarkable success in various machine learning applications, often outperforming other techniques in terms of predictive accuracy. However, their performance can be influenced by factors such as the choice of weak learners, the number of iterations, and the hyperparameter settings (e.g., learning rate, tree depth, regularization parameters). Appropriate tuning and selection of the boosting algorithm and its hyperparameters are crucial for achieving optimal performance on specific tasks and datasets.

4 Application of ML in Predicting the Strength of FRP-Reinforced Concrete Elements

Accurately predicting strength parameters, such as compressive and shear strength, is crucial for designing and assessing fiber-reinforced polymer (FRP)-reinforced concrete structures. However, the intricate interactions between the various influencing factors, including concrete mix design, FRP composite properties, interface bond behavior, and loading conditions, pose significant challenges for conventional empirical models to capture the complex nonlinearities inherent in these structural systems.

In recent years, ML techniques have emerged as promising data-driven alternatives, offering the potential to model the strength behavior of FRP-reinforced concrete elements with improved accuracy and reliability. By leveraging the ability of ML algorithms to learn from large datasets and identify intricate patterns, these techniques can effectively capture the complex relationships between input parameters and target strength properties.

This section provides a comprehensive review of the state-of-the-art application of ML approaches to predict the compressive and shear strength of FRP-reinforced concrete elements. It covers a wide range of ML techniques, including artificial neural networks (ANNs), support vector machines (SVMs), genetic programming (GP), ensemble learning models (e.g., random forests, gradient boosting), and emerging techniques such as gene expression programming (GEP) and multigene genetic programming (MGP). The review analyzes these ML models' strengths, limitations, and predictive capabilities, highlighting their performance compared to conventional empirical models and design guidelines.

Furthermore, the review examines the influence of various input parameters, including concrete properties, FRP reinforcement characteristics, geometric configurations, and loading conditions, on the accuracy and robustness of the ML models. It also discusses the challenges associated with data availability, model optimization, and reliability assessment of ML techniques in the context of strength prediction for FRP-reinforced concrete structures.

By synthesizing the latest research advances and identifying current gaps and future research directions, this section aims to provide a comprehensive understanding of the potential and limitations of ML techniques in this domain, ultimately contributing to the development of more accurate and reliable strength prediction models for FRP-reinforced concrete elements.

4.1 Compressive Strength

The prediction of compressive strength in fiber-reinforced polymer (FRP)-confined concrete elements has been an active area of research due to the significant enhancement in load-bearing capacity and ductility provided by the external confinement. Conventional design approaches based on empirical equations often fail to accurately capture the intricate interactions between the various parameters influencing the compressive behavior. Consequently, ML techniques have emerged as promising data-driven alternatives for modeling the compressive strength of FRP-confined concrete elements.

To provide a comprehensive overview of the reviewed studies, Table 1 summarizes the salient characteristics of the various ML approaches employed for predicting the compressive strength of FRP-confined concrete elements. The tabulated information includes the reference, ML technique utilized, dataset size, column cross-section, number of input parameters, and a list of the specific input variables considered, facilitating a convenient comparison of the different ML models, their scope, and the input parameters accounted for in each study.

Among the pioneering studies exploring the application of ML techniques for predicting the compressive strength of FRP-confined concrete elements, Cevik and Guzelbey [54] and Naderpour et al. [55] employed artificial neural networks (ANNs) to model the strength enhancement of concrete cylinders confined with carbon fibre-reinforced polymer (CFRP). Their results demonstrated the potential of ANNs to capture the intricate interrelationships among the various input parameters governing the compressive behavior, outperforming existing empirical models. Elsanadedy et al. (2012) further solidified the advantage of ANNs over regression models in predicting the compressive strength and crushing strain of FRP-confined concrete cylinders.

Subsequent investigations explored the application of diverse ML algorithms to this challenge. Jalal and Ramezanianpour [56] derived an explicit strength prediction equation from an optimized ANN model, providing a simplified yet accurate model for design applications. Pham and Hadi [57] proposed ANN models tailored for predicting the compressive strength and strain of FRP-confined square/

Table 1 Summary of ML techniques for predicting the compressive strength of FRP confined concrete columns

N°	Ref	Year	ML type	Data size	Column cross-section	No. of input	Input variables
1	[54]	2008	ANN	101	Circular	4	d, t_f, E_f, f'_{c0}
2	[55]	2010	ANN	213	Circular	6	$d, h, f'_{c0}, t_f, f_f, E_f$
3	[65]	2012	ANN	272	Circular	5	$d, t_f, E_f, f_f, f'_{c0}$
4	[56]	2012	ANN	128	Circular	6	$d, h, t_f, E_f, \varepsilon_{h,rup}, f'_{c0}$
5	[57]	2014	ANN	104	Square / Rectangular	8	$b, h, r, f'_{c0}, \varepsilon_{c0}, t_f, E_f, f_f$
6	[58]	2015	Fuzzy logic	100	Square / Rectangular	5	b, h, t_f, f'_{c0}, E_f
7	[59]	2016	GP	832	Circular	5	$f'_{c0}, d, t_f, E_f, \varepsilon_{h,rup}$
8	[60]	2018	ANN, ANFIS	341	Circular	4	f'_{c0}, d, t_f, f_f
9	[66]	2017	SVR	238	Circular	5	$d, h/d, f'_{c0}, t_f, f_f$
10	[67]	2017	ANN	465	Circular	4	f'_{c0}, E_f, t_f, d
11	[68]	2020	ANN	708	Circular	5	d, h, t_f, E_f, f'_{c0}
12	[69]	2020	ANN	708	Circular	5	d, h, t_f, E_f, f'_{c0}
13	[61]	2020	ANN	169	Circular	8	$d, h, f'_{c0}, \varepsilon_{c0}, E_c, t_f, E_f, \varepsilon_{h,rup}$
14	[62]	2020	ANN	281	Circular	6	$d, h, f'_{c0}, t_f, f_l, f_f$
15	[63]	2022	GMDH	221	Circular	11	$d, h, f'_{c0}, \varepsilon_{c0}, E_c, t_f, E_f, \varepsilon_{h,rup}, f_l, \rho_k, \rho_\varepsilon$

(continued)

Table 1 (continued)

N°	Ref	Year	ML type	Data size	Column cross-section	No. of input	Input variables
16	[64]	2022	GEP	828	Circular	5	d, h, t_f, E_f, f'_{c0}
17	[70]	2023	XGB, CatB, RF, LGBM, Extra trees, KNN, SVR	675	Circular	6	f'_{c0}, d, h, f_f, E_f, t_f
18	[71]	2023	MARS, ELM, MARS–PSO, ELM–PSO	281	Circular	6	d, h, f'_{c0}, t_f, f_l, f_f
19	[72]	2023	ANN-PSO, ANN-GA, ANN-CBO, ANN-ECBO	223	Circular	6	f'_{c0}, d, h, f_f, E_f, t_f
20	[73]	2023	LR, RR, Lasso, PRM, DT, ANN	416	Square / Rectangular	7	b, h, r, f'_{c0}, t_f, E_f, f_f
21	[74]	2023	ANN, GPR, SVR	1151	Circular	11	f'_{c0}, d, h, f'_{c0}, F_{ty}, n, b_f, t_f, O_f, E_f, f_f
22	[75]	2024	PSO-CatB, CatB, XGB, AdaB, GBM, Extra Trees, RF	916	Circular	8	d, h, f'_{c0}, ρ_f, t_f, E_f, f_f, n

Designations: GMDH = Group Method of Data Handling, MARS = Multivariate Adaptive Regression Spline, ELM = Extreme Learning Machine, PSO = Particle Swarm Optimization, GA = Genetic Algorithm, CBO = Colliding Bodies Optimization, ECBO = Enhanced Colliding Bodies Optimization, LR = Linear Regression, RR = Ridge Regression, Lasso = Lasso Regression, PRM = Polynomial Regression, GPR = Gaussian process regression, d = cylinder diameter; t_f = FRP thickness; E_f = FRP elastic modulus; $f'_{c0=}$ unconfined compressive strength; h = cylinder height; f_f = FRP tensile strength; $\varepsilon_{h,rup}$ = hoop rupture strain of FRP; b = short side length; h = long side length; r = corner radius; ε_{c0} = peak strain of unconfined concrete; h/d = cylinder height/cylinder diameter; E_c = elastic modulus of concrete; f_l = lateral confining pressure; ρ_k = stiffness ratio; ρ_ε = strain ratio; F_{ty} = FRP type; n = number of FRP layers; b_f = FRP width; O_f = orientation of FRP; ρ_f = FRP reinforcement ratio.

rectangular concrete columns, achieving superior performance compared to existing methods. Doran et al. [58] developed a fuzzy logic-based model to predict the lateral confinement coefficient of CFRP-wrapped concrete columns, demonstrating improved predictive performance compared to non-linear regression models.

Lim et al. [59] pioneered the use of genetic programming (GP) to develop closed-form expressions for predicting the ultimate condition of FRP-confined concrete, including compressive strength, ultimate axial strain, and hoop rupture strain, outperforming existing models in terms of accuracy and precision. Mansouri et al. [60] investigated various artificial intelligence techniques, such as ANNs, radial basis neural networks (RBNNs), and adaptive neuro-fuzzy inference systems (ANFIS), for estimating the peak and residual conditions of actively confined concrete, with the RBNN exhibiting the lowest prediction errors.

Jiang et al. [61] introduced a data-driven neural network method for predicting the ultimate conditions [61] and stress–strain response of FRP-confined concrete, outperforming traditional design-oriented mathematical models in terms of accuracy and adaptability. Kamgar et al. [62] employed a feed-forward backpropagation neural network (FFBPNN) combined with k-fold cross-validation to develop an explicit formulation for estimating the compressive strength of FRP-confined concrete cylinders, achieving superior accuracy compared to existing empirical models.

Deng et al. [63] utilized the Group Method of Data Handling (GMDH) to establish a confinement model that outperformed existing models, particularly for predicting the ultimate axial strain of FRP-confined concrete, a crucial parameter for design considerations. Ilyas et al. [64] explored gene expression programming (GEP) to predict the compressive strength of CFRP-confined circular concrete columns, demonstrating its superiority over previous approaches in terms of accuracy and performance metrics.

More recent contributions have focused on evaluating and comparing the performance of various ML algorithms for this task. Cakiroglu [70] investigated gradient boosting techniques (XGBoost, CatBoost), random forests, and support vector regression (SVR) for predicting the mechanical properties of aramid FRP (AFRP)-confined concrete, with XGBoost, CatBoost, and k-nearest neighbors (KNN) performing best, further enhanced by data augmentation using generative adversarial networks (GANs).

Hanteh et al. [71] combined multivariate adaptive regression splines (MARS) and extreme learning machines (ELMs) with particle swarm optimization (PSO) to estimate the compressive strength of FRP-confined concrete columns, with the MARS-PSO model achieving the highest accuracy among the evaluated models. Kaveh and Khavaninzadeh [72] optimized and compared various ANN structures using metaheuristic algorithms, with the ECBO-FFB model (enhanced CBO-FFB) exhibiting the lowest prediction errors and highest accuracy.

Khodadadi Koodiani et al. [73] explored multiple ML algorithms, including linear regression, decision trees, and ANNs, for predicting the compressive strength of FRP-confined square/rectangular concrete columns, with the ANN model providing the most accurate predictions, and parameter dependencies were analyzed using

techniques like partial dependence plots and contour plots. Kumar et al. [74] developed and compared several ML models, including ANNs, Gaussian process regression (GPR), and support vector machines (SVMs), with the optimized GPR model achieving the best performance in predicting the compressive strength of FRP-confined concrete cylinders.

Most recently, Khodadadi et al. [75] introduced a novel PSO-CatBoost model, combining particle swarm optimization (PSO) and the CatBoost algorithm, for predicting the compressive strength of CFRP-confined circular concrete specimens, achieving superior performance compared to other ML models and empirical/analytical approaches, and identifying the compressive strength of unconfined concrete and CFRP reinforcement ratio as the most critical predictors.

The extensive body of research in this domain has showcased the potential of ML techniques to accurately predict the compressive strength of FRP-confined concrete elements. The diverse range of ML approaches employed, including ANNs, SVR, fuzzy logic, GEP, GMDH, ensemble learning and hybrid models, have demonstrated their ability to capture the complex interactions and nonlinearities inherent in the compressive behavior of these structural systems. These ML models have consistently outperformed conventional empirical equations and design guidelines, offering increased accuracy, reliability, and computational efficiency.

The reviewed studies reflect the continuous evolution and exploration of different ML approaches to better capture the complex compressive behavior of FRP-confined concrete elements. Furthermore, the variation in dataset sizes, ranging from relatively smaller (e.g., 100 data points) to larger datasets exceeding 800 data points, highlights the efforts to enhance the generalizability and robustness of the ML models, albeit with challenges in data acquisition, preprocessing, and computational requirements. In addition to the prevalent circular column cross-sections, some works have also encompassed square/rectangular configurations, providing a more comprehensive representation of different structural elements. The number of input parameters considered in the ML models varies significantly, from as few as four to as many as eleven, with a larger number of inputs potentially capturing more intricate interactions but also increasing model complexity and the risk of overfitting for smaller datasets. While certain input parameters, such as column geometry, unconfined concrete strength, FRP thickness, and FRP material properties, are consistently included, some researchers have explored the inclusion of additional parameters, such as concrete elastic modulus, strain properties, and confinement characteristics, further enriching the input space and potentially improving model performance.

4.2 Shear Strength

Accurately predicting the shear strength in reinforced concrete (RC) beams strengthened with externally bonded fiber-reinforced polymer (EB-FRP) has posed significant challenges due to the complexity of the underlying shear failure mechanisms. Conventional empirical equations have struggled to capture the intricate interactions

between the various components that contribute effectively to the shear capacity. Consequently, researchers have turned to ML techniques as data-driven alternatives, offering the potential to model the shear behavior of EB FRP-reinforced concrete elements with improved accuracy and reliability.

Table 2 summarizes the key characteristics of various ML approaches employed in this domain, including the reference, technique, dataset size, FRP configuration, number of input parameters, and specific input variables considered. This overview facilitates a comparative analysis of the different ML models, their scopes, and the factors accounted for in each study.

Among the pioneering studies exploring the application of ML techniques for predicting the shear strength of EB FRP-reinforced RC beams, Perera et al. [76, 77] employed artificial neural networks (ANNs) and genetic algorithms (GAs). Their research focused on estimating the shear capacity of RC beams strengthened with U-shaped and wrapped FRP configurations. The results demonstrated the potential of these ML techniques to capture the intricate interrelationships among the various input parameters governing the shear behavior, outperforming conventional empirical models. Concurrently, Nehdi and Nikopour [78] utilized GAs to optimize shear design equations for RC beams reinforced with EB-FRP sheets, accounting for the interaction mechanisms between concrete, steel reinforcement, and FRP laminates. Tanarslan et al. [79] developed an ANN model that incorporated the influence of the shear span-to-depth ratio, a crucial parameter often overlooked in design guidelines, thereby enhancing the model's predictive capabilities.

Subsequent investigations explored the application of diverse ML algorithms to this challenge. Naderpour and Alavi [80] introduced an adaptive neuro-fuzzy inference system (ANFIS) to estimate the shear contribution of EB-FRP in RC beams, demonstrating superior performance compared to existing guideline equations. Kar et al. [83] and Kar and Biswal [84] employed ANFIS models to predict the shear contribution of EB-FRP in wrapped and U-wrapped RC beams, respectively, highlighting the capability of ANFIS to consider the combined effect of multiple input parameters. Kamgar et al. [81] employed multigene genetic programming (MGP) to formulate a new equation for predicting the shear capacity of EB FRP-strengthened RC beams, outperforming conventional methods in terms of accuracy and precision. Abuodeh et al. [82] applied resilient back-propagating neural networks (RBPNNs) in conjunction with recursive feature elimination (RFE) and neural interpretation diagrams (NIDs) to identify the critical parameters influencing the shear capacity of side-bonded and U-wrapped EB FRP-reinforced RC beams. Karzad et al. [85] developed an ANN model to account for the potential adverse interaction between EB-FRP and steel stirrups, an important consideration in the design and assessment of these structural systems. Gasser et al. [88] proposed an ANN model for estimating the shear strength of EB FRP-strengthened RC beams, considering various parameters such as element geometry, strengthening details, and configurations, demonstrating better agreement with experimental data compared to previous models. Taghipour Anvari et al. [86] utilized gene expression programming (GEP) to develop data-driven models for predicting the total shear strength of RC beams strengthened with

Table 2 Summary of ML techniques for predicting the shear strength of EB FRP reinforced concrete beams

N°	Ref	Year	ML type	Data size	Column cross-section	No. of input	Input variables
1	[76]	2010	ANN, GA	46	U	9	b_w, h_b, ρ_f, α_f, E_f, ρ_{sl}, A_{90}, f_{ys}, f_{ck}
2	[77]	2010	ANN	98	U, W	11	b_w, h_b, ρ_f, α_f, E_f, ρ_{sl}, A_{90}, f_{ys}, f_{ck}, a/d_e, *type of configuration*
3	[78]	2010	GA	212	SB, U, W	6	a/d_e, f_{ck}, b_w, d_e, ε_{fu}, Γ_f
4	[79]	2012	ANN	84	U, W	9	b_w, d_e, f_{ck}, w_f/s_f, α_f, E_f, ε_{fu}, t_f, a/d_e
5	[80]	2017	ANFIS	89	U, W	6	ε_{fu}, d_f, f_{ck}, a/d_e, B, R
6	[81]	2020	MGP	89	U, W	6	ε_{fu}, d_f, f_{ck}, a/d_e, B, R
7	[82]	2020	ANN with RFE and NID	120	SB, U	12	b_w, d_e, a/d_e, f_{ck}, f_{ys}, A_v/S_v, t_f, h_f, w_f/s_f, B_f, E_f, f_f
8	83]	2020	ANFIS	119	W	11	d_e, a, f_{ck}, A_{sv}, w_f, s_f, α_f, n, t_f, E_f, ε_{fu}
9	[84]	2020	ANFIS	148	U	12	b_w, a/d_e, f_{ck}, A_{sv}, d_f, w_f, s_f, α_f, n, t_f, E_f, ε_{fu}

(continued)

Table 2 (continued)

N°	Ref	Year	ML type	Data size	Column cross-section	No. of input	Input variables
10	[85]	2022	ANN	191	SB, U	16	f_{ck}, a/d_e, ρ_{sl}, f_{yl}, S_v, ρ_s, f_{ys}, E_f, ε_{fu}, h_f, w_f, s_f, t_f, *type of stirrups, FRP type, FRP distribution*
11	[86]	2023	GEP	785	SB, U, W	9	d_e, a/d_e, f_{ck}, b_w, f_{ys}, E_f, ρ_f, ρ_s, ε_{fu}
12	[87]	2023	XGB, RF, AdaB, ElasticNet, KNN, GBM	196	SB, U, W	15	b_w, d_e, f_{ck}, f_{ys}, A_{sv}, S_v, a/d_e, E_f, ε_{fu}, f_f, t_f, w_f, s_f, h_f, α_f
13	[88]	2023	ANN	200	SB, U, W	13	b_w, h_b, d_e, h_f, w_f, t_f, s_f, ρ_f, α_f, E_f, f_f, ε_{fu}, *wrapping scheme*
14	[89]	2023	ANN, XGB, RF, CatB, LGBM, AdaB	315	SB, U, W	8	a/d_e, ρ_s, h_f, ρ_f, *RM, FM, CS*, α_f
15	[90]	. 2023	LR, RR, LASSO, KNN, SVR, DT, RF, XGB, AdaB, CatB	584	SB, U, W	12	b_w, d_e, a/d_e, h_f, E_f, ε_{fu}, A_{sv}, f_{ck}, t_f, f_f, *type of wrapping scheme, type of fiber*
16	[91]	2024	RF, XGB, AdaB, CatB	302	SB, U, W	12	b_w, d_e, a/d_e, h_f, E_f, ε_{fu}, A_{sv}, f_{ck}, t_f, f_f, *type of wrapping scheme, type of fiber*

Designations: U = U-shaped FRP; W = fully wrapped FRP; SB = side-bonded FRP; b_w = beam width; h_b = beam height; ρ_f = ratio of the FRP transversal reinforcement; α_f = FRP angle of inclination; E_f = FRP elastic modulus; ρ_{sl} = ratio of the longitudinal steel reinforcement; A_{90} = cross-sectional area of transverse steel per length unit; f_{ys} = design yielding stress of the shear steel reinforcement; f_{ck} = concrete compressive strength; a/d_e = shear span to depth ratio; d_e = beam effective depth; ε_{fu} = FRP ultimate strain; w_f/s_f = FRP width-spacing ratio; t_f = FRP thickness; d_f = FRP effective depth; f_{ys} = design yielding stress of the shear steel reinforcement; A_v/S_v = area of stirrups-to-spacing ratio; h_f = FRP sheet height; B_f = FRP width; f_f = FRP tensile strength; a = shear span; A_{sv} = internal shear reinforcement ratio; w_f = width of FRP strips; s_f = center to center spacing of FRP strips; n = number of layers of FRP; f_{yl} = design yielding stress of the longitudinal steel reinforcement; S_v = stirrups spacing; ρ_s = ratio of the shear steel reinforcement; *RM* = reinforcement method; *FM* = failure mode; *CS* = cross-section of RC beam.

EB-FRP sheets, revealing the significant influence of the RC beam depth and the effective depth of the EB-FRP sheets on the total shear strength.

More recent contributions have focused on evaluating and comparing the performance of various ML algorithms for this task. Ezami et al. [87] conducted a comprehensive evaluation of algorithms such as XGBoost, Random Forest (RF), AdaBoost, ElasticNet, K-Nearest Neighbors (KNN), and Gradient Boosting (GB), identifying XGBoost as the superior performer in terms of accuracy and reliability. Wang et al. [89] employed interpretable ML techniques, including ANNs, XGBoost, RF, CatBoost, LightGBM, and AdaBoost, to predict the contribution of EB-FRP to the shear strength of RC beams with internal shear reinforcement. Their research derived an explicit equation based on the XGBoost model, combining ML accuracy with physical meaning. Rahman et al. [90] and Rahman and Muntasir Billah [91] developed multiple ML models, encompassing Linear Regression (LR), Ridge Regression (RR), Lasso Regression (LASSO), KNN, Support Vector Regression (SVR), Decision Tree (DT), Random Forest (RF), XGBoost, AdaBoost, and CatBoost, to predict the shear strength of EB FRP-reinforced RC beams. Their studies demonstrated the superior performance of ensemble learning models, particularly RF, CatBoost, and XGBoost, in estimating the shear capacity of both rectangular and T-beams, outperforming existing mechanics-driven models. Additionally, they employed Shapley Additive exPlanations (SHAP) to interpret the importance of input features, revealing the significant influence of parameters such as the effective depth and the height of the EB-FRP layers.

The reviewed studies collectively illustrate the potential of ML techniques to enhance the accuracy and reliability of shear strength predictions for EB FRP-reinforced concrete elements. A wide range of ML approaches, including ANNs, GAs, ANFIS, MGP, SVR, decision trees, and ensemble learning models, have been successfully employed, consistently outperforming traditional empirical equations and design guidelines.

While early studies focused predominantly on U-shaped and wrapped FRP configurations, recent investigations have expanded to incorporate side-bonded schemes, providing a more comprehensive representation of practical reinforcement strategies. The diversity in dataset sizes, ranging from relatively small (e.g., 46 data points) to large (exceeding 500 data points), underscores the ongoing efforts to improve model generalizability and robustness, albeit with inherent challenges in data acquisition and computational demands. The number of input parameters considered in these ML models varies considerably, from as few as six to as many as sixteen, reflecting the complexity of the shear behavior and the trade-offs between capturing intricate interactions and the risk of overfitting, especially for smaller datasets. While fundamental factors such as beam geometry, concrete strength, shear steel reinforcement, and FRP reinforcement details are consistently included, researchers have also explored the incorporation of additional parameters, such as longitudinal steel reinforcement details, stirrup types, fiber types, and FRP distribution patterns, potentially enhancing the predictive capabilities of the ML models.

4.3 Discussion and Recommendations

The comprehensive review of machine learning (ML) techniques for predicting the compressive and shear strength of FRP-reinforced concrete elements has highlighted the significant progress made in this domain, as well as the challenges and opportunities for further research and development.

Firstly, the studies have consistently demonstrated the superior predictive capabilities of ML models compared to conventional empirical equations and design guidelines. By leveraging advanced algorithms and computational power, ML techniques can effectively capture the complex nonlinearities and interactions between multiple influencing parameters, resulting in improved accuracy and reliability in strength predictions. This is particularly evident in the case of compressive strength prediction for FRP-confined concrete elements, where ML models have consistently outperformed traditional design-oriented models.

Secondly, the diversity of ML techniques explored in this domain is noteworthy, reflecting the continuous evolution and exploration of different approaches to better capture the intricate behavior of FRP-reinforced concrete elements. From early applications of artificial neural networks (ANNs) and genetic algorithms (GAs) to more recent advancements in ensemble learning models (e.g., random forests, gradient boosting), and multigene genetic programming (MGP), and hybrid models, the field has witnessed a continuous expansion and refinement of ML methodologies tailored for strength prediction tasks.

However, several challenges remain to be addressed to facilitate the practical adoption and widespread implementation of ML techniques in the design and assessment of FRP-reinforced concrete structures.

- Data Availability and Quality: The development of accurate and reliable ML models hinges on the availability of comprehensive and diverse experimental databases, encompassing a wide range of material properties, geometric configurations, loading conditions, and failure modes. Larger and more representative datasets are essential for training and validating the ML models, ensuring their generalizability and robustness. Efforts should be directed towards standardizing data collection and documentation practices, enabling collaborative data sharing and model development across research groups and institutions.
- Model Interpretability and Physical Consistency: While data-driven ML approaches have demonstrated superior predictive performance, practitioners often seek to bridge the gap between these solutions and traditional mechanistic approaches. Enhancing the interpretability and physical consistency of ML models is crucial for their acceptance and integration into existing design frameworks. Efforts should be made to develop interpretable ML models that can provide insights into the underlying physical mechanisms, potentially through hybrid approaches that combine data-driven techniques with mechanistic models or by incorporating physics-informed constraints and domain knowledge into the ML algorithms.

- Uncertainty Quantification and Reliability Assessment: The reliability assessment and uncertainty quantification of ML models in the context of strength predictions for FRP-reinforced concrete structures warrant further investigation. Methodologies for quantifying model uncertainties, accounting for potential outliers or anomalies in the data, and establishing confidence intervals are crucial for practical implementation and acceptance by the engineering community. Techniques such as Bayesian neural networks, conformal prediction, and ensemble methods can be explored to enhance the reliability and trustworthiness of ML-based strength predictions.
- Transfer Learning and Domain Adaptation: In scenarios where data availability is limited or restricted to specific domains, transfer learning and domain adaptation techniques can be leveraged to transfer knowledge from related domains or datasets, potentially reducing the data requirements for developing accurate ML models. These techniques can facilitate the efficient development of ML models for new material systems, geometric configurations, or loading conditions, leveraging knowledge from previously trained models or related domains.
- Integration with Structural Analysis and Design Frameworks: Ultimately, the true potential of ML-based strength prediction models can be realized through their seamless integration with structural analysis and design frameworks. By coupling these data-driven models with finite element analysis, performance-based design methodologies, and decision support systems, engineers can leverage the strengths of ML techniques to optimize the design and assessment of FRP-reinforced concrete structures, considering multiple performance criteria and design constraints.

By addressing these challenges and leveraging the synergies between data-driven ML techniques and traditional engineering principles, the field of structural engineering can unlock the full potential of these advanced modeling approaches. Collaborative efforts among researchers, practitioners, and industry stakeholders are essential to overcome the remaining hurdles and pave the way for more efficient, reliable, and economical solutions to design and assess FRP-reinforced concrete structures.

5 Conclusion

The application of machine learning (ML) techniques has emerged as a promising approach for accurately predicting the compressive and shear strengths of externally bonded fiber-reinforced polymer (EB FRP)-reinforced concrete elements. This comprehensive review has critically analyzed the state-of-the-art ML methodologies employed in this domain, encompassing a diverse array of techniques, including artificial neural networks (ANNs), support vector machines (SVMs), genetic programming (GP), ensemble learning models, and emerging algorithms such as gene expression programming (GEP) and multigene genetic programming (MGP).

The reviewed studies have consistently demonstrated the superior predictive capabilities of ML models in capturing the complex nonlinearities and interactions among the various influencing parameters, outperforming conventional empirical equations and design guidelines. By leveraging advanced computational power and data-driven modeling approaches, ML techniques have effectively addressed the limitations of traditional mechanistic models, offering improved accuracy, reliability, and adaptability in strength predictions for FRP-reinforced concrete structures.

Despite the significant progress made in this field, several challenges remain to be addressed to facilitate the practical adoption and widespread implementation of ML techniques in the design and assessment of FRP-reinforced concrete elements. These challenges include ensuring data availability and quality, enhancing model interpretability and physical consistency, quantifying uncertainties and assessing reliability, leveraging transfer learning and domain adaptation techniques, and seamlessly integrating ML-based strength prediction models with structural analysis and design frameworks.

Collaborative efforts among researchers, practitioners, and industry stakeholders are crucial to overcome these remaining hurdles. By combining the strengths of data-driven ML techniques with traditional engineering principles and domain knowledge, the field of structural engineering can unlock the full potential of these advanced modeling approaches, paving the way for more efficient, reliable, and economical solutions in the design and assessment of FRP-reinforced concrete structures.

As the adoption of FRP composites continues to grow in the construction industry, the development of accurate and reliable strength prediction models will play a pivotal role in ensuring the safe and cost-effective implementation of these innovative materials. The fusion of ML methodologies and structural engineering expertise holds immense promise in advancing the state-of-the-art in this domain, ultimately contributing to the realization of more sustainable, resilient, and high-performance concrete structures.

References

1. Sen, R., Mullins, G.: Application of FRP composites for underwater piles repair. Compos. B Eng. **38**, 751–758 (2007)
2. Zhou, Y., Guo, M., Sui, L., Xing, F., Hu, B., Huang, Z., Yun, Y.: Shear strength components of adjustable hybrid bonded CFRP shear-strengthened RC beams. Compos. B Eng. **163**, 36–51 (2019)
3. Chen, W., Pham, T.M., Sichembe, H., Chen, L., Hao, H.: Experimental study of flexural behaviour of RC beams strengthened by longitudinal and U-shaped basalt FRP sheet. Compos. B Eng. **134**, 114–126 (2018)
4. Michel, L., Ferrier, E., Agbossou, A., Hamelin, P.: Flexural stiffness modelling of RC slab strengthened by externally bonded FRP. Compos. B Eng. **40**, 758–765 (2009)
5. Oller, E., Pujol, M., Marí, A.: Contribution of externally bonded FRP shear reinforcement to the shear strength of RC beams. Compos. B Eng. **164**, 235–248 (2019)
6. Van Den Einde, L., Zhao, L., Seible, F.: Use of FRP composites in civil structural applications. Constr. Build. Mater. **17**, 389–403 (2003)

7. Rousakis, T.C., Karabinis, A.I.: Adequately FRP confined reinforced concrete columns under axial compressive monotonic or cyclic loading. Materials and Structures/Materiaux et Constructions. **45**, 957–975 (2012)
8. Wu, Y.F., Jiang, J.F.: Effective strain of FRP for confined circular concrete columns. Compos. Struct. **95**, 479–491 (2013)
9. Chen, G.M., Teng, J.G., Chen, J.F., Rosenboom, O.A.: Interaction between Steel Stirrups and Shear-Strengthening FRP Strips in RC Beams. J. Compos. Constr. **14**, 498–509 (2010)
10. Khalifa, A., Nanni, A.: Improving shear capacity of existing RC T-section beams using CFRP composites. Cem. Concr. Compos. **22**, 165–174 (2000)
11. Oller, E., Kotynia, R., Marí, A.: Assessment of the existing models to evaluate the shear strength contribution of externally bonded frp shear reinforcements. Compos. Struct. **266**, 113641 (2021)
12. Huang, X., Zhou, Y., Li, W., Hu, B., Zhang, J.: Reliability-based design of FRP shear strengthened reinforced concrete Beams: Guidelines assessment and calibration. Compos. Struct. **323**, 117421 (2023)
13. Adeli, H., Yeh, C.: Perceptron Learning in Engineering Design. Computer-Aided Civil and Infrastructure Engineering. **4**, 247–256 (1989)
14. Kaveh, A., Gholipour, Y., Rahami, H.: Optimal Design of Transmission Towers Using Genetic Algorithm and Neural Networks. **23**, 1–19 (2008)
15. Cha, Y.J., Choi, W., Büyüköztürk, O.: Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. Computer-Aided Civil and Infrastructure Engineering. **32**, 361–378 (2017)
16. Hoskere, V., Narazaki, Y., Hoang, T.A., Spencer, B.F.: MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure. J Civ Struct Health Monit. **10**, 757–773 (2020)
17. Alkayem, N.F., Cao, M., Zhang, Y., Bayat, M., Su, Z.: Structural damage detection using finite element model updating with evolutionary algorithms: a survey. Neural Comput. Appl. **30**, 389–411 (2018)
18. Bini, S.A.: Artificial Intelligence, Machine Learning, Deep Learning, and Cognitive Computing: What Do These Terms Mean and How Will They Impact Health Care? J. Arthroplasty **33**, 2358–2361 (2018)
19. Dimiduk, D.M., Holm, E.A., Niezgoda, S.R.: Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering. Integrating Materials and Manufacturing Innovation 2018 7:3. 7, 157–172 (2018).
20. Ongsulee, P.: Artificial intelligence, machine learning and deep learning. International Conference on ICT and Knowledge Engineering. 1–6 (2018).
21. Gianey, H.K., Choudhary, R.: Comprehensive Review On Supervised Machine Learning Algorithms. Proceedings - 2017 International Conference on Machine Learning and Data Science, MLDS 2017. 2018-January, 38–43 (2017).
22. Amanpreet Singh, Narina Thakur, Aakanksha Sharma: A review of supervised machine learning algorithms | IEEE Conference Publication | IEEE Xplore.
23. Shang, M., Li, H., Ahmad, A., Ahmad, W., Ostrowski, K.A., Aslam, F., Joyklad, P., Majka, T.M.: Predicting the Mechanical Properties of RCA-Based Concrete Using Supervised Machine Learning Algorithms. Materials 2022, Vol. 15, Page 647. 15, 647 (2022).
24. Ahmad, A., Ahmad, W., Aslam, F., Joyklad, P.: Compressive strength prediction of fly ash-based geopolymer concrete via advanced machine learning techniques. Case Studies in Construction Materials. **16**, e00840 (2022)
25. Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., Aljaaf, A.J.: A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. 3–21 (2020).
26. Cao, M., Alkayem, N.F., Pan, L., Novák, D., Cao, M., Alkayem, N.F., Pan, L., Novák, D.: Advanced Methods in Neural Networks-Based Sensitivity Analysis with their Applications in Civil Engineering. Artificial Neural Networks - Models and Applications. (2016).
27. Roh, Y., Heo, G., Whang, S.E.: A Survey on Data Collection for Machine Learning: A Big Data-AI Integration Perspective. IEEE Trans. Knowl. Data Eng. **33**, 1328–1347 (2021)

28. Tapeh, A.T.G., Naser, M.Z.: Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices. Archives of Computational Methods in Engineering 2022 30:1. 30, 115–159 (2022).
29. Nithurshan, M., Elakneswaran, Y.: A systematic review and assessment of concrete strength prediction models. Case Studies in Construction Materials. **18**, e01830 (2023)
30. Alkayem, N.F., Shen, L., Mayya, A., Asteris, P.G., Fu, R., Di Luzio, G., Strauss, A., Cao, M.: Prediction of concrete and FRC properties at high temperature using machine and deep learning: A review of recent advances and future perspectives. Journal of Building Engineering. **83**, 108369 (2024)
31. Alexandropoulos, E., Papoutsidakis, M., Nikitakos, N.: SCADA Backup System for the Control of Networked Valves in Modern Ships Facilities. Int. J. Comput. Appl. **178**, 975–8887 (2019)
32. Ayesha, S., Hanif, M.K., Talib, R.: Overview and comparative study of dimensionality reduction techniques for high dimensional data. Information Fusion. **59**, 44–58 (2020)
33. Ben Chaabene, W., Flah, M., Nehdi, M.L.: Machine learning prediction of mechanical properties of concrete: Critical review. Constr. Build. Mater. **260**, 119889 (2020)
34. Naser, M.Z., Alavi, A.H.: Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. Architecture, Structures and Construction 2021 3:4. 3, 499–517 (2021).
35. Bardhan, A., Asteris, P.G.: Application of hybrid ANN paradigms built with nature inspired meta-heuristics for modelling soil compaction parameters. Transportation Geotechnics. **41**, 100995 (2023)
36. Géron, A.: Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow (2019, O'reilly). Hands-On Machine Learning with R. 510 (2017).
37. Rosa, J.L.G., Rosa, J.L.G.: Artificial Neural Networks - Models and Applications. Artificial Neural Networks - Models and Applications. (2016).
38. Abunassar, N., Alas, M., Ali, S.I.A.: Prediction of Compressive Strength in Self-compacting Concrete Containing Fly Ash and Silica Fume Using ANN and SVM. Arab. J. Sci. Eng. **48**, 5171–5184 (2023)
39. Bi, Y., Xue, B., Zhang, M.: Evolutionary Computation and Genetic Programming. Adapt. Learn. Optim. **24**, 49–74 (2021)
40. Zhong, J., Feng, L., Ong, Y.S.: Gene Expression Programming: A Survey [Review Article], (2017).
41. Salleh, M.N.M., Talpur, N., Hussain, K.: Adaptive Neuro-Fuzzy Inference System: Overview, Strengths, Limitations, and Solutions. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 10387 LNCS, 527–535 (2017).
42. Boser, B.E., Guyon, I.M., Vapnik, V.N.: Training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory. 144–152 (1992).
43. Cortes, C., Vapnik, V., Saitta, L.: Support-vector networks. Machine Learning 1995 20:3. 20, 273–297 (1995).
44. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V., Critianini, N., Shawe-Taylor, J., Williamson, B.: Support Vector Clustering. J. Mach. Learn. Res. **2**, 125–137 (2001)
45. Tharwat, A.: Parameter investigation of support vector machine classifier with kernel functions. Knowl. Inf. Syst. **61**, 1269–1302 (2019)
46. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
47. Schapire, R.E.: A Brief Introduction to Boosting. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. pp. 1401–1406. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999).
48. Freund, Y., Schapire, R.: Experiments with a New Boosting Algorithm. International Conference on Machine Learning. (1996).
49. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. **55**, 119–139 (1997)

50. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. **29**, 1189–1232 (2001)
51. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 13–17-August-2016, 785–794 (2016).
52. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: Guyon, I., Luxburg, U. Von, Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.) Advances in Neural Information Processing Systems. Curran Associates, Inc. (2017).
53. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: unbiased boosting with categorical features. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.) Advances in Neural Information Processing Systems. Curran Associates, Inc. (2018).
54. Cevik, A., Guzelbey, I.H.: Neural network modeling of strength enhancement for CFRP confined concrete cylinders. Build. Environ. **43**, 751–763 (2008)
55. Naderpour, H., Kheyroddin, A., Amiri, G.G.: Prediction of FRP-confined compressive strength of concrete using artificial neural networks. Compos. Struct. **92**, 2817–2829 (2010)
56. Jalal, M., Ramezanianpour, A.A.: Strength enhancement modeling of concrete cylinders confined with CFRP composites using artificial neural networks. Compos. B Eng. **43**, 2990–3000 (2012)
57. Pham, T.M., Hadi, M.N.S.: Predicting Stress and Strain of FRP-Confined Square/Rectangular Columns Using Artificial Neural Networks. Journal of Composites for Construction. 18, (2014).
58. Doran, B., Yetilmezsoy, K., Murtazaoglu, S.: Application of fuzzy logic approach in predicting the lateral confinement coefficient for RC columns wrapped with CFRP. Eng. Struct. **88**, 74–91 (2015)
59. Lim, J.C., Karakus, M., Ozbakkaloglu, T.: Evaluation of ultimate conditions of FRP-confined concrete columns using genetic programming. Comput. Struct. **162**, 28–37 (2016)
60. Mansouri, I., Gholampour, A., Kisi, O., Ozbakkaloglu, T.: Evaluation of peak and residual conditions of actively confined concrete using neuro-fuzzy and neural computing techniques. Neural Comput. Appl. **29**, 873–888 (2018)
61. Jiang, K., Han, Q., Bai, Y., Du, X.: Data-driven ultimate conditions prediction and stress-strain model for FRP-confined concrete. Compos Struct. 242, (2020).
62. Kamgar, R., Naderpour, H., Komeleh, H.E., Jakubczyk-Gałczyńska, A., Jankowski, R.: A proposed soft computing model for ultimate strength estimation of FRP-confined concrete cylinders. Applied Sciences (Switzerland). 10, (2020).
63. Deng, C., Zhang, R., Xue, X.: Application of Group Method of Data Handling on the Ultimate Conditions' Prediction of FRP-Confined Concrete Cylinders. Polymers (Basel). 14, (2022).
64. Ilyas, I., Zafar, A., Afzal, M.T., Javed, M.F., Alrowais, R., Althoey, F., Mohamed, A.M., Mohamed, A., Vatin, N.I.: Advanced Machine Learning Modeling Approach for Prediction of Compressive Strength of FRP Confined Concrete Using Multiphysics Genetic Expression Programming. Polymers (Basel). 14, (2022).
65. Elsanadedy, H.M., Al-Salloum, Y.A., Abbas, H., Alsayed, S.H.: Prediction of strength parameters of FRP-confined concrete. Compos. B Eng. **43**, 228–239 (2012)
66. Mozumder, R.A., Roy, B., Laskar, A.I.: Support Vector Regression Approach to Predict the Strength of FRP Confined Concrete. Arab. J. Sci. Eng. **42**, 1129–1146 (2017)
67. Cascardi, A., Micelli, F., Aiello, M.A.: An Artificial Neural Networks model for the prediction of the compressive strength of FRP-confined concrete circular columns. Eng. Struct. **140**, 199–208 (2017)
68. Ahmad, A., Plevris, V., Khan, Q.U.Z.: Prediction of properties of frp-confined concrete cylinders based on artificial neural networks. Crystals (Basel). **10**, 1–22 (2020)
69. Ahmad, A., Khan, Q. uz Z., Raza, A.: Reliability analysis of strength models for CFRP-confined concrete cylinders. Compos Struct. 244, (2020).
70. Cakiroglu, C.: Explainable Data-Driven Ensemble Learning Models for the Mechanical Properties Prediction of Concrete Confined by Aramid Fiber-Reinforced Polymer Wraps Using Generative Adversarial Networks. Appl. Sci. **13**, 11991 (2023)

71. Hanteh, M., Malek, H., Kheyroddin, A.: Proposing the combined MARS–PSO and ELM–PSO models for estimating the compressive strength of circular concrete columns wrapped with FRP sheets. Soft comput. **27**, 15937–15953 (2023)
72. Kaveh, A., Khavaninzadeh, N.: Efficient training of two ANNs using four meta-heuristic algorithms for predicting the FRP strength. Structures. **52**, 256–272 (2023)
73. Khodadadi Koodiani, H., Erfanian, N., Majlesi, A., Hosseinzadeh, A., Jafari, E., Shahin, M., Matamoros, A.: Calibrating equations to predict the compressive strength of FRP-Confined columns using optimized neural network model. Structures. 56, (2023).
74. Kumar, P., Arora, H.C., Bahrami, A., Kumar, A., Kumar, K.: Development of a Reliable Machine Learning Model to Predict Compressive Strength of FRP-Confined Concrete Cylinders. Buildings. 13, (2023).
75. Khodadadi, N., Roghani, H., De Caso, F., El-kenawy, E.S.M., Yesha, Y., Nanni, A.: Data-driven PSO-CatBoost machine learning model to predict the compressive strength of CFRP- confined circular concrete specimens. Thin-Walled Structures. 198, (2024).
76. Perera, R., Arteaga, A., Diego, A.D.: Artificial intelligence techniques for prediction of the capacity of RC beams strengthened in shear with external FRP reinforcement. Compos. Struct. **92**, 1169–1175 (2010)
77. Perera, R., Barchín, M., Arteaga, A., Diego, A.D.: Prediction of the ultimate strength of reinforced concrete beams FRP-strengthened in shear using neural networks. Compos. B Eng. **41**, 287–298 (2010)
78. Nehdi, M., Nikopour, H.: Genetic algorithm model for shear capacity of RC beams reinforced with externally bonded FRP. Materials and Structures/Materiaux et Constructions. **44**, 1249–1258 (2011)
79. Tanarslan, H.M., Secer, M., Kumanlioglu, A.: An approach for estimating the capacity of RC beams strengthened in shear with FRP reinforcements using artificial neural networks. Constr. Build. Mater. **30**, 556–568 (2012)
80. Naderpour, H., Alavi, S.A.: A proposed model to estimate shear contribution of FRP in strengthened RC beams in terms of Adaptive Neuro-Fuzzy Inference System. Compos. Struct. **170**, 215–227 (2017)
81. Kamgar, R., Bagherinejad, M.H., Heidarzadeh, H.: A new formulation for prediction of the shear capacity of FRP in strengthened reinforced concrete beams. Soft comput. **24**, 6871–6887 (2020)
82. Abuodeh, O.R., Abdalla, J.A., Hawileh, R.A.: Prediction of shear strength and behavior of RC beams strengthened with externally bonded FRP sheets using machine learning techniques. Compos Struct. 234, (2020).
83. Kar, S., Pandit, A.R., Biswal, K.C.: Prediction of FRP shear contribution for wrapped shear deficient RC beams using adaptive neuro-fuzzy inference system (ANFIS). Structures. **23**, 702–717 (2020)
84. Kar, S., Biswal, K.C.: FRP shear contribution prediction for U-wrapped RC T-beams using a soft computing tool. Structures. **27**, 1093–1104 (2020)
85. Karzad, A.S., Khalil, M.A., Mohamed, A.M., Barakat, S.: Modeling the shear capacity of externally bonded fiber reinforced polymer strengthened beams by artificial neural network. International Journal of Applied Science and Engineering. 20, (2023).
86. Taghipour Anvari, A., Babanajad, S., Gandomi, A.H.: Data-Driven Prediction Models For Total Shear Strength of Reinforced Concrete Beams With Fiber Reinforced Polymers Using An Evolutionary Machine Learning Approach. Eng Struct. 276, (2023).
87. Ezami, N., Özyüksel Çiftçioğlu, A., Mirrashid, M., Naderpour, H.: Advancing Shear Capacity Estimation in Rectangular RC Beams: A Cutting-Edge Artificial Intelligence Approach for Assessing the Contribution of FRP. Sustainability. **15**, 16126 (2023)
88. Gasser, M., Mahmoud, O., Elsayed, T., Deifalla, A.: Reliable machine learning for the shear strength of beams strengthened using externally bonded FRP jackets. Front Mater. 10, (2023).
89. Wang, C., Zou, X., Sneed, L.H., Zhang, F., Zheng, K., Xu, H., Li, G.: Shear strength prediction of FRP-strengthened concrete beams using interpretable machine learning. Constr Build Mater. 407, (2023).

90. Rahman, J., Arafin, P., Muntasir Billah, A.H.M.: Machine learning models for predicting concrete beams shear strength externally bonded with FRP. Structures. 53, 514–536 (2023).
91. Rahman, J., Muntasir Billah, A.H.M.: Interpretable Ensemble Machine Learning Models for Shear Strength Prediction of Reinforced Concrete Beams Externally Bonded with FRP. In: Lecture Notes in Civil Engineering. pp. 1265–1278. Springer Science and Business Media Deutschland GmbH (2024).

A Modified Jaya Algorithm for Optimum Design of Carbon Fiber-Reinforced Polymers

Sinan Melih Nigdeli, Gebrail Bekdaş, and Aylin Ece Kayabekir

Abstract In this study, a modified Jaya algorithm is presented for a structural engineering problem. Jaya algorithm (JA) is a single-phase metaheuristic algorithm using best and worst existing solutions. The effects of these solutions are equally considered in the classical form. In the proposed modified algorithm, a weight factor is considered for these factors. This factor is randomly chosen to keep the algorithm as a parameter-free one. The results of modified JA (MJA) and JA are compared and evaluated for optimization of carbon fiber-polymers (CFRP) used for increasing the shear force capacity of reinforced concrete beams. The results show that MJA outperforms JA.

Keywords Jaya algorithm · Optimization · CFRP · Modified jaya algorithm · Shear force · Reinforced concrete

1 Introduction

In civil engineering, the structural component design frequently draws inspiration from nature and utilizes metaheuristic algorithms. The principal aim of these design processes is to reduce material expenses, which are intricately linked to the overall weight of the structure. Crafting an economically efficient design that complies with numerous constraints can render the engineering challenge highly non-linear. In

S. M. Nigdeli · G. Bekdaş (✉)
Department of Civil Engineering, Istanbul University—Cerrahpaşa, Avcılar, 34320 Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

A. E. Kayabekir
Department of Civil Engineering, Istanbul Gelisim University, Avcılar, 34310 Istanbul, Turkey
e-mail: aekayabekir@gelisim.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_13

such scenarios, the most practical approach involves the application of numerical iterations.

Optimization in structural engineering aims to achieve a well-balanced solution that balances safety and cost-effectiveness. In pursuit of this goal, an objective is minimized or maximized while adhering to the specified design constraints. Metaheuristic methods, inspired by natural and life phenomena, serve as algorithms that have found applications as numerical optimization tools in various structural engineering problems. These include weight optimization of trusses [1–3], tuning of tuned mass dampers [4–6], optimizing the design of reinforced concrete (RC) members [7–9] and optimizing the design of RC retaining walls [10–12].

In reinforced concrete (RC) structures, shear reinforcements, commonly known as stirrups, are employed to counteract shear forces. When retrofitting or expanding existing structures due to seismic considerations, it may become necessary to enhance their shear capacity. One approach to achieving this is by introducing additional stirrups, which may entail the jacketing of existing structural members. In such cases, the cross-sectional area of the RC element is increased, impacting both the overall weight and rigidity of the structure. Consequently, the retrofitted design must be treated as a new project and undergo a comprehensive analysis. This method represents a means of augmenting shear force capacity.

Another option is to implement carbon fiber-reinforced polymer (CFRP) materials in the seismic retrofit process. When compared to the first option, this approach maintains the existing structure's rigidity and weight while improving shear capacity. CFRP is advantageous due to its ease of application and its linear deformation behavior, coupled with a significant strain capacity. Metaheuristics were also used in optimizing the design of carbon fiber-reinforced polymer (CFRP) components [13, 14].

In this research, a modified version of the Jaya algorithm is introduced to address a specific structural engineering problem. The Jaya algorithm (JA) is a single-phase metaheuristic algorithm that relies on the best and worst existing solutions, giving equal consideration to their effects in its conventional form. In the proposed modified algorithm, a weight factor is introduced for these factors. This factor is randomly selected, making the algorithm parameter-free. The study involves a comparison and evaluation of the results obtained using the modified JA (MJA) and the conventional JA for the optimization of carbon fiber-reinforced polymers (CFRP) utilized to enhance the shear force capacity of reinforced concrete beams. The findings indicate that MJA surpasses JA in terms of performance.

2 Modified Jaya Algorithm

The Jaya algorithm (JA) [15] is a simple optimization technique that aims to find the best solution to a problem. It works by continuously improving a set of potential solutions. During each iteration, it identifies the best and worst solutions in the current set and adjusts the solutions to get closer to the best one while introducing randomness

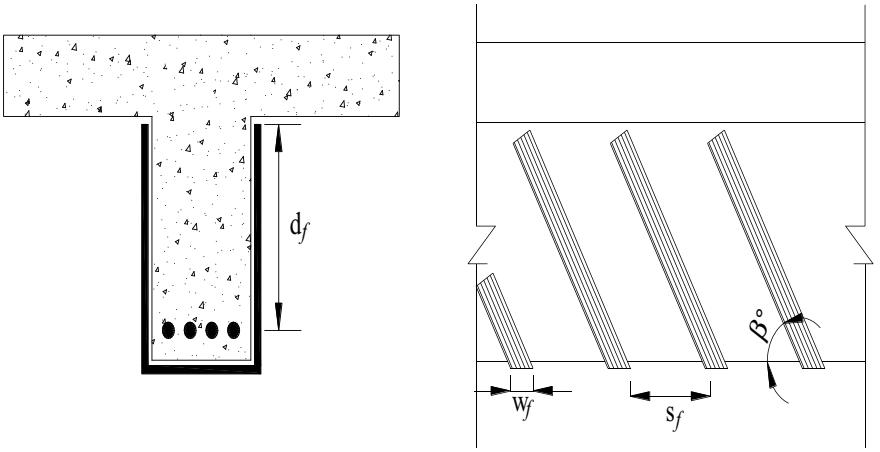


Fig. 1 The design problem

to explore new possibilities. This process continues until a stopping condition is met. The Jaya algorithm doesn't require many parameters, making it straightforward to use.

In the context of iterative optimization, a candidate solution for a design variable (represented as x_i^{t+1}) within a population (where i ranges from 1 to N) is determined based on the current solution (x_i^t), two random numbers selected from the range of 0 to 1 (denoted as r_1 and r_2), and the best (g*) and worst (g_w) design variables currently present within the population, as specified in Eq. (1).

$$x_i^{t+1} = x_i^t + r_1\left(g^* - |x_i^t|\right) - r_2\left(g^w - |x_i^t|\right) \tag{1}$$

In the modified JA (MJA), the weights are considered as r_1 and $(1\text{-}r_1)$ for the best and worst solutions. In that case, the sum of the weights of considering best and worst is equal to 1. The formulation of MFA is given as Eq. (2).

$$x_i^{t+1} = x_i^t + r_1\left(g^* - |x_i^t|\right) - (1 - r_1)\left(g^w - |x_i^t|\right) \tag{2}$$

3 The Optimum CFRP Design

The objective of the optimization is to reduce the necessary CFRP area per meter (denoted as A), aiming to enhance the shear force capacity of a reinforced concrete (RC) beam. The design parameters include the width (w_f), spacing (s_f), and angle (β) of CFRP, with specified ranges provided in Table 1. The constant values, as well

as the variable dimensions, are illustrated in Fig. 1 depicting the beam featuring a T-shaped cross-section. The depth of the RC member covered by CFRP is represented as d_f, formulated by Eq. 3 when the height of the RC slab is expressed as h_f. The equation for the area of CFRP strips is provided in Eq. 3, and in accordance with ACI 318 [16], the three design constants are specified as Eqs. (4–7).

$$d_f = d - h_f \tag{3}$$

$$A = \frac{w_f\left(\frac{2d_f}{\sin\beta} + b\right)}{s_f} x1000 \tag{4}$$

$$g_1(x) : s_f \leq {}^{d}/_{4} \tag{5}$$

$$g_2(x) : V_{additional} < 0.7R\frac{(2t_f w_f f_{fe})(\sin\beta + \cos\beta)d_f}{s_f + w_f} \tag{6}$$

$$g_3(x) : \frac{(2t_f w_f f_{fe})(\sin\beta + \cos\beta)}{s_f + w_f} \leq \frac{2\sqrt{f_c'}b_w d}{3} - V_s \tag{7}$$

4 Numerical Examples

The optimization process comprises 10,000 iterations, repeated across 20 runs, with a population number set at 20. Design constraints violations result in the objective function being penalized at a rate of 10^{16} mm^2 per meter. Optimal results are outlined in Table 2. These tables include the corresponding values for JA and MJA. For comparative analysis, the best (Abest) and mean (Amean) CFRP area, along with standard deviation and the iteration count required to achieve optimal results, are also presented.

5 Conclusion

The optimized design variables for JA and MJA showcase distinct differences. Notably, the optimal width (w_f) for MJA is considerably larger at 85.51 mm compared to the more modest 43.14 mm in JA. Similarly, the spacing (s_f) for MJA is larger at 30.42 mm, contrasting with the smaller 15.36 mm in JA. The angle (β) sees an increase in MJA, reaching 65.19° compared to 65.14° in JA. Despite these variations, the best CFRP area (Abest) remains almost identical between JA (716,352.84 mm^2) and MJA (716,354.20 mm^2). However, the mean CFRP area (Amean) is higher for MJA (716,630.70 mm^2) compared to JA (716,523.16 mm^2). Notably, MJA exhibits

Table 1 The design constants

Constant	Symbol	Unit	Value
Breadth	b_w	mm	200
Height	h	mm	500
Effective depth	d	mm	450
Thickness of CFRP	t_f	mm	0.165
Reduction factor	R	-	0.5
Thickness of slab	h_f	mm	100
Comp. strength of concrete	f_c'	MPa	20
Effective tensile strength of CFRP	f_{fe}	MPa	3790
Width of CFRP	w_f	mm	10–100
Specific of CFRP	s_f	mm	0-d/4
Angle of CFRP	β	°	0–90
Additional shear force	$V_{additional}$	kN	150
Shear force capacity of rebar	V_s	kN	50

Table 2 The optimum results

	JA	MJA
w_f (mm)	43.14	85.51
s_f (mm)	15.36	30.42
β (°)	65.14	65.19
A_{best} (mm^2)	716,352.84	716,354.20
A_{mean} (mm^2)	716,523.16	716,630.70
Standard deviation	549.3010	251.2024
Iteration number	8487	7832

a significantly lower standard deviation (251.2024) compared to JA (549.3010), suggesting a more stable optimization process. Furthermore, MJA achieves the optimal solution in fewer iterations (7,832) compared to JA (8,487). In summary, while the best CFRP area remains close for both cases, the revised results indicate differences in design variables, mean CFRP area, standard deviation, and iteration numbers between JA and MJA.

References

1. Kanarachos, S., Griffin, J., Fitzpatrick, M.E.: Efficient truss optimization using the contrast-based fruit fly optimization algorithm. Comput. Struct. **182**, 137–148 (2017)
2. Degertekin, S.O., Lamberti, L., Ugur, I.B.: Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. Appl. Soft Comput. **70**, 903928 (2018)

3. Baghlani, A., Makiabadi, M.H., Maheri, M.R.: Sizing optimization of truss structures by an efficient constraint-handling strategy in TLBO. J. Comput. Civ. Eng. **31**(4), 04017004 (2017)
4. Bekdaş, G., Nigdeli, S.M.: Mass ratio factor for optimum tuned mass damper strategies. Int. J. Mech. Sci. **71**, 68–84 (2013)
5. Salvi, J., Rizzi, E.: Optimum tuning of Tuned Mass Dampers for frame structures under earthquake excitation. Struct. Control. Health Monit. **22**(4), 707–725 (2015)
6. Nigdeli, S.M., Bekdaş, G.: Optimum tuned mass damper design in frequency domain for structures. KSCE J. Civ. Eng. **21**(3), 912–922 (2017)
7. Nigdeli, S.M., Bekdas, G., Kim, S., Geem, Z.W.: A novel harmony search based optimization of reinforced concrete biaxially loaded columns. Struct. Eng. Mech. **54**(6), 1097–1109 (2015)
8. Govindaraj, V., Ramasamy, J.V.: Optimum detailed design of reinforced concrete continuous beams using genetic algorithms. Comput. Struct. **84**(1–2), 34–48 (2005)
9. Lee, C., Ahn, J.: Flexural design of reinforced concrete frames by genetic algorithm. J. Struct. Eng. **129**(6), 762774 (2003)
10. Ceranic, B., Fryer, C., Baines, R.W.: An application of simulated annealing to the optimum design of reinforced concrete retaining structures. Comput. Struct. **79**(17), 1569–1581 (2001)
11. Yepes, V., Gonzalez-Vidosa, F., Alcala, J., Villalba, P.: CO 2-optimization design of reinforced concrete retaining walls based on a VNS-threshold acceptance strategy. J. Comput. Civ. Eng. **26**(3), 378386 (2011)
12. Yepes, V., Alcala, J., Perea, C., GonzálezVidosa, F.: A parametric study of optimum earth-retaining walls by simulated annealing. Eng. Struct. **30**(3), 821830 (2008)
13. Kayabekir A.E., Sayin B., Bekdas G., Nigdeli S. M. Optimum carbon fiber reinforced polymer design for increasing shear capacity of RC beams. In: 3rd International Conference on Engineering and Natural Sciences (ICENS 2017), 3–7 May 2017, Budapest, Hungary
14. Kayabekir, A. E., Sayın, B., Nigdeli, S. M., Bekdaş, G. (2018). Jaya algorithm based optimum carbon fiber reinforced polymer design for reinforced concrete beams. In: AIP Conference Proceedings (Vol. 1978, No. 1, p. 260006). AIP Publishing
15. Rao, R.: Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. **7**(1), 19–34 (2016)
16. ACI 318M-05, Building code requirements for structural concrete and commentary. American Concrete Institute (2005)

Prediction of Bi-Linear Strength Envelope of Brazilian Soils Using Machine Learning Techniques

Jonathan do Amaral Braz, Leonardo Goliatt da Fonseca, and Tatiana Tavares Rodriguez

Abstract The knowledge of soil strength envelope is essential for several works in civil engineering, but laboratory tests to obtain them can be costly. However, based on recently produced studies, it is believed that the development of computational models to estimate them is a tool capable of meeting this demand. This study aims, therefore, to develop a machine learning model capable of estimating bi-linear strength envelopes of soils. In order to achieve it, a database was built with results from characterization, direct shear and Brazilian tests from Brazilian soils. Five regression algorithms were applied to the database, undergoing training, cross-validation, hyperparameter optimization and testing phases. Most models presented satisfactory performance metrics, emphasizing Linear Regression algorithm as the most adequate for direct shear tests and Extremely Randomized Trees algorithm for Brazilian tests.

Keywords Machine learning · Regression analysis · Soils · Strength envelope

1 Introduction

Soil strength is a property of great importance for solving several engineering problems, as it is specially relevant in cases of ground failure. Some projects in which this information is required are foundation design, retaining wall design, slope stability analysis and the mapping of urban areas at risk of landslides [1]. Such property can be expressed by strength parameters, which can be obtained from strength envelopes

J. do Amaral Braz (✉) · L. G. da Fonseca · T. T. Rodriguez
José Lourenço Kelmer N/A, Federal University of Juiz de Fora, Juiz de Fora 36036-900, Brazil
e-mail: jonathan.amaral@engenharia.ufjf.br

L. G. da Fonseca
e-mail: leonardo.goliatt@ufjf.edu.br

T. T. Rodriguez
e-mail: tatiana.rodriguez@ufjf.edu.br

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_14

that demonstrate the relation between normal stresses and shear stresses in a soil mass. These envelopes can be attained through laboratory tests, such as direct shear, triaxial compression [2] and tensile strength compression [3]. However, performing these tests can be costly, both in terms of time and in terms of required equipment and skilled work.

Machine learning techniques have been developed throughout the last decades and, in recent years, have been gaining prominence in literature. The ability to teach an algorithm to predict outputs consistent with the data provided is a very relevant tool in several fields of knowledge. In civil engineering, it can be specially useful in situations of no data available or no tools for performing tests. Literature presents several works in this field, such as estimating soil water retention curves based on their physical characterization [4] and predicting soil classification based on their load capacity [5, 6].

This study aims to present the development of a model capable of estimating a bi-linear strength envelope for saturated and unsaturated soils, as another application of machine learning techniques on civil engineering problems. To this end, a database of laboratory tests was built, from which features were selected and applied to five machine learning algorithms. Then, such algorithms underwent hyperparameter optimization to increase the models' accuracy, in order to select the most appropriate one for this problem. The chapter is divided into five parts, being the first one this introduction. The second part presents fundamental concepts about soil strength and the ways of obtaining such property. The third part discusses the machine learning algorithms that were used in this study and how they work to predict data. In the fourth one, a database built with results from the presented tests is applied on the algorithms. Finally, the fifth part presents conclusions on the performed work.

2 Soil Strength Envelopes

Soil masses are structures naturally exposed to stresses, either from self-weight, external loads or inertial forces caused by seismic and other vibrational motions [1]. Such stresses result in reactions within the soil mass. They are organized into normal and shear stress. It can be said, therefore, that geological materials have an intrinsic ability to resist these external stresses. In geotechnical engineering, such ability is called soil strength [7].

Such strength can vary according to the amount of water contained in the sample. Soils in a saturated condition have lower resistance when compared to unsaturated soils. This happens because unsaturated soils rely on the phenomenon of water retention, which maintains a higher cohesion between particles [8].

Soils load capacity can be expressed graphically by the relation between the normal stresses acting on a mass and its ability to resist shear stresses. Based on this, it is possible to establish an analysis of a stress state that causes soil failure. These graphical representations are called strength envelopes. There are multiple modulations of envelopes, the most common being Mohr's—characterized by a curve

generated from rupture circles—and Coulomb's—characterized by a straight line. As Mohr's envelopes are difficult to apply, it is common to approximate them in a straight line, as in Coulomb's envelopes. Widely used in geotechnics, this technique is known as the Mohr–Coulomb criterion [2].

The resistance envelope can be defined as the locus that represents the stress state in a soil mass. This means that shear stresses with values above the curve result in the failure of the soil mass. Likewise, shear stresses with values that touch the curve indicate a limit equilibrium and, lastly, stresses below the curve indicate the stability of the soil [2].

To obtain a strength envelope, there are several tests that can provide a relation between normal stresses and shear stresses. The most common are the triaxial compression test and the direct shear test. In the latter, a specimen is placed in a shear box. Then, a normal stress and a tangential stress are applied in the upper part of the box, causing the displacement of the specimen upper part, leading to its failure [9].

In some cases, a linear strength envelope can generate inadequate results in its applications, such as overestimated safety factors in slope stability analyses [10]. In light of that, a bi-linear envelope can then be more representative. One of the ways of obtaining this kind of envelope is by inserting data from tensile strength tests into the graphical representation. Proposed by Lobo Carneiro and Barcelos [11] and also known as the Brazilian test, it determines tensile strength indirectly by compressing the side of a cylindrical specimen with a uniform load until the specimen exhibits cracking on one of its faces.

Finally, it is possible to extract strength parameters from the strength envelopes. Such parameters are the cohesive intercept and the friction angle [2]. These parameters, in turn, are used in calculation methodologies in engineering problems, such as the ones mentioned above.

3 Machine Learning Algorithms

Machine learning can broadly be defined as the ability of an algorithm to learn quality from data provided. In computational terms, it is the improvement of a task based on experience, followed by the ability to estimate results yet unknown to the machine with satisfactory performance metrics [12].

Literature presents a range of different algorithms, from simpler models—such as Linear Regression, that adjusts samples to a linear function—to more complex models—such as neural networks, that have structures with several layers and non-linear learning techniques [12]. As mentioned above, there are several possible applications in many fields. Especially interesting here are the studies on laboratory test results in engineering, such as estimating binder content in asphalt mixtures [13] and predicting strength parameters for solid urban waste [14].

The problems that can be solved through machine learning can be organized into three categories: classification, regression and clustering. The application case to be presented later in this chapter is a regression analysis, which consists in finding the

relation between input variables and a numerical output variable [15]. The algorithms used have different levels of complexity and are briefly explained below.

3.1 Linear Regression

Linear regression is one of the simplest statistical and computational methodologies. It consists of predicting a dependent variable based on the weighted sum of independent variables [12]. This relation can be expressed in generalized form by Eq. 1:

$$\nu = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \tag{1}$$

where:

$\hat{y}$ is the predicted value;
n is the number of independent variables;
x_i is the ith independent variable;
θ_j is the jth parameter of the model.

The algorithm aims to find the straight line that best fits a scatter plot of the samples. The criterion for the fit quality is the distance between the line and the samples, which should be as small as possible. One of the most commonly used fitting methods for this model is the least squares method [15].

3.2 Support Vector Regressor

Support Vector Regressor (SVR) is a variation from Support Vector Machine (SVM), a model initially proposed by Cortes and Vapnik [16] to solve classification problems. This algorithm aims to find the function that best fits the data provided and is suitable for dealing with both linear and non-linear fits.

SVR has a set of hyperparameters that allow its specificity. The ε hyperparameter acts as a margin around the regression function, trying to include as many samples as possible in its range and penalizing outliers. The hyperparameter C has the function of regularization, trying to find a balance between the amplitude of margins and the sensitivity to distant points. A fit with wide margins, which includes as many samples as possible, can be insensitive to outliers and consequently compromise the model's performance [12].

Finally, there are the kernels. These hyperparameters are functions that establish non-linear relations between the independent variables and the dependent variable. Consequently, this allows data to be distributed in multidimensional spaces, so that the model can find linear separations between the variables and perform more complex analyses [12].

3.3 Decision Tree Regressor

Like Support Vector Machine, this model was initially developed to solve classification problems, but is also suitable for regression problems [12]. Proposed by Breiman et al. [17], decision trees are a group of high-performance algorithms. The structure of decision trees starts from a single point, the root, which is successively divided into branches. At the end of each branch is a decision point, called node. At the end of the tree as a whole, there are points that do not have decision questions but contain the final results of the analysis. These are called leaves.

The process of training a decision tree consists in splitting data into two groups at each branch. The aim of this division is to create a binary question that can reduce the level of dispersion between the data. The algorithm then goes through the entire system, answering these questions based on the training stage, in an attempt to find the appropriate value [12].

It is important to note that decision trees, despite their high performance, can be more susceptible to overfitting. This happens because of their high flexibility during the training stage. In the process of dividing up the data, the algorithm tries to establish a criterion that organizes samples into groups that are as homogeneous as possible and, as a consequence, it can result in a model that fits data from the training phase appropriately, but cannot be generalized.

3.4 Random Forest Regressor

Also proposed by Breiman [18], Random Forest Regressor is a form of ensemble algorithms. This approach consists of combining results generated by individual models to obtain a more robust final answer. In the case of Random Forest Regressor, the elements grouped together are decision trees.

It is known that random trees can be prone to overfitting due to their high capacity for variation, which can result in biased criteria during training. In the case of random forests, to avoid this problem, each of the trees receives a different slice of the database. These slices are generated randomly using two techniques. The first one, called bagging, consists in the possibility of repeated entries during the generation of the sample, and the second one, called pasting, does not consider this repetition [12].

In a random forest, when a slice is generated at each tree node, several possible divisions are considered for each of the sample characteristics. The final split selected is the one with the least entropy in each split group. At the end of the process, the outputs of each decision tree are compared and a final result, based on what was presented by the majority, is selected as the model output [12].

3.5 Extremely Randomized Trees Regressor

Also known as Extra Trees Regressor, the model was proposed by Geurst, Ernst and Whenkel [19]. It works as an extension of Random Forest Regressor, differentiating itself by focusing on increasing randomness during its operation, resulting in more robust and generally more accurate models compared to decision trees.

As previously mentioned, random forests go through a meticulous process when splitting data, in order to reduce entropy in the generated groups. On one hand, this feature allows the algorithm to provide more accurate results. On the other hand, it increases the risk of overfitting the model. In contrast to this feature, Extra Trees Regressor introduces an additional layer of randomness during the decision trees construction. During the node division stage, the algorithm has no specific criteria for organizing the splitted groups. This feature increases the model's ability to generalize, as well as reducing its training time, which is a costly stage.

Extra Trees is an appropriate choice for problems where computational efficiency is crucial or when there is a large amount of data to be processed.

4 Applying Direct Shear and Brazilian Tests Data on Regression Algorithms

Having presented both the engineering problems that require costly laboratory data and the possibility of using machine learning algorithms to estimate certain information, this section aims to show a possible application: the development of a machine learning model to predict the bi-linear strength envelope of soils based on their physical characteristics and on results from the two tests described above.

4.1 Database Description

The database was built using information on sample classification—gravel, sand, silt and clay content—, Atterberg limits and specific gravity values, associated with test parameters and results from direct shear tests [20–28] and Brazilian tests [3, 8, 26–34].

Table 1 shows a slice of the database for the direct shear tests, containing the physical characterization information (as described above), the normal stresses applied during the tests and the respective shear stresses obtained. Table 2, in turn, shows a slice of the database for the Brazilian tests, also with physical characterization of the samples and their respective tensile strengths.

Table 1 Database slice for direct shear tests

	Sample #10	Sample #15	Sample #111	Sample #118
Gravel content (%)	0.0	0.0	1.0	0.0
Sand content (%)	21.0	30.0	20.0	40.0
Silt content (%)	27.0	62.0	8.0	20.0
Clay content (%)	52.0	8.0	71.0	40.0
Plastic limit (%)	34.0	26.0	39.0	30.0
Liquid limit (%)	59.0	46.0	87.0	55.0
Specific gravity (g/cm^3)	2.80	2.70	2.69	2.66
Normal stress (kPa)	12.50	25.00	50.00	100.00
Shear stress (kPa)	40.53	49.69	70.20	61.60

Table 2 Database slice for Brazilian tests

	Sample #1	Sample #14	Sample #201	Sample #38
Gravel content (%)	1.0	0.0	0.0	1.0
Sand content (%)	33.0	21.0	30.0	25.0
Silt content (%)	5.0	27.0	62.0	9.0
Clay content (%)	62.0	52.0	8.0	64.0
Plastic limit (%)	31.1	34.0	26.0	36.0
Liquid limit (%)	57.7	59.0	46.0	49.6
Specific gravity (g/cm^3)	2.82	2.80	2.70	2.79
Tensile stress (kPa)	42.400	3.277	0.000	28.705

4.2 Data Processing

The database contains 179 samples for direct shear tests and 154 samples for Brazilian tests. During data collection, it was detected that some information regarding physical characterization was missing. Due to the relatively low number of samples for both cases, it was decided to fill in the missing data with the average value for each category.

It is also important to comprehend the relation between the variables and the final tensile and shear strength values. For that reason, a correlation analysis was carried out using the Spearman method. At the end of the analysis, it was decided to select the independent variables with the highest absolute correlation values to be used as features. The dependent variables selected are "Shear stress (kPa)" for the direct shear test and "Tensile stress (kPa)" for the Brazilian test. Table 3 shows the correlation values between the independent variables and shear stress. Table 4 shows the correlation between tensile strength and the other variables. For direct shear tests, the variables selected as features were "Normal stress (kPa)", "Specific gravity (g/cm^3)", "Gravel content (%)" and "Liquid limit (%)". For Brazilian tests,

Table 3 Correlation values for direct shear tests features

	Shear stress (kPa)
Normal stress (kPa)	0.861453
Specific gravity (g/cm^3)	0.173592
Gravel content (%)	0.138437
Liquid limit (%)	-0.081007
Clay content (%)	-0.074240
Silt content (%)	0.066638
Plastic limit (%)	-0.061191
Sand content (%)	0.048778

Table 4 Correlation values for Brazilian tests features

	Tensile stress (kPa)
Gravel content (%)	-0.267922
Liquid limit (%)	0.217774
Silt content (%)	0.187126
Specific gravity (g/cm^3)	-0.134809
Sand content (%)	-0.084589
Clay content (%)	0.072008
Plastic limit (%)	-0.050642

the variables selected were "Gravel content (%)", "Liquid limit (%)", "Silt content (%)" and "Specific gravity (g/cm^3)".

During feature processing, data normalization was conducted, in order to facilitate the training process. Normalization consists in defining variable values within a range from 0 to 1 without altering the original relation between them. Also, the database was divided into two sets: 70% of the samples were splitted for the training and cross-validation stages, while 30% were for testing phase. The cross-validation method chosen for this study was k-fold with 10 folds, in order to reduce bias.

To develop the model, Python's scikit-learn library was used, which contains functions for all the machine learning algorithms discussed in the third section of this chapter. The decision of which algorithm to use in the work was based on a benchmark, which consists of applying the data to the models and comparing their performance metrics to find out which one gives the most satisfactory results. Within this study, the performance metric used was R^2.

In each function, it is possible to adjust the hyperparameters of each algorithm. Initially, it was decided to keep the default parameters for each of them in the first selection stage and apply the grid search for optimization parameters, this being the second stage.

Table 5 Algorithm benchmark for direct shear tests

Algorithm	R^2	Standard deviation
Linear Regression	0.82	0.12
Support Vector Regressor	-0.22	0.19
Decision Tree Regressor	0.73	0.23
Random Forest Regressor	0.80	0.17
Extra Trees Regressor	0.79	0.19

4.3 Algorithm Training, Validation, Hyperparameter Optimization and Testing

Table 5 shows the results of the first phase for the shear test. The Linear Regression algorithm performed best, with an R^2 of 0.82. The Support Vector Regressor algorithm showed the worst performance among all of them, with an R^2 of -0.22. The Decision Tree Regressor algorithm had an R^2 of 0.73 and the decision tree-based models performed slightly better: 0.80 for Random Forest Regressor and 0.79 for Extra Trees Regressor. Regarding the ensemble algorithms, metrics are consistent with the theory of higher forest performance, although they failed to show better values than Linear Regression.

Table 6 shows the results of the first stage for the shear test. Support Vector Regressor showed the worst performance metrics, with an R^2 of -0.30, followed by the Linear Regression algorithm, which showed an R^2 of 0.14. The algorithms based on decision trees, on the other hand, showed more satisfactory performances: 0.76 for the Decision Tree Regressor, 0.78 for the Extra Trees Regressor and 0.82 for the Random Forest Regressor, the latter being the algorithm with the best performance.

Next, the algorithms underwent a hyperparameter optimization, which were described in Sect. 3. This process was done using a grid search, which consists of a search method that tests all specified combinations of hyperparameters to determine the best model configuration in order to present the most satisfactory performance. Table 7 shows a comparison between scikit-learn's default values and the best values found for the direct shear tests. Table 8 shows the same comparison for Brazilian tests.

The algorithms were then tested with hyperparameter values obtained from the grid search. Table 9 shows the results for the direct shear tests. In general, there

Table 6 Algorithm benchmark for Brazilian tests

Algorithm	R^2	Standard deviation
Linear Regression	0.14	0.19
Support Vector Regressor	-0.30	0.10
Decision Tree Regressor	0.76	0.20
Random Forest Regressor	0.82	0.17
Extra Trees Regressor	0.78	0.19

Table 7 Hyperparameters best values for direct shear tests

Algorithm	Hyperparameter	Standard value	Best value
Support Vector Regressor	C	1.0	10.0
	epsilon	0.1	1.0
Decision Tree Regressor	max_depth	None	None
	min_samples_split	2	2
	min_samples_leaf	1	4
Random Forest Regressor	n_estimators	100	100
	max_depth	None	10
	min_samples_split	2	10
	min_samples_leaf	1	2
Extra Trees Regressor	n_estimators	100	300
	max_depth	None	10
	min_samples_split	2	10
	min_samples_leaf	1	2

Table 8 Hyperparameters best values for Brazilian tests

Algorithm	Hyperparameter	Standard value	Best value
Support Vector Regressor	C	1.0	10.0
	epsilon	0.1	1.0
Decision Tree Regressor	max_depth	None	None
	min_samples_split	2	1
	min_samples_leaf	1	2
Random Forest Regressor	n_estimators	100	100
	max_depth	None	20
	min_samples_split	2	2
	min_samples_leaf	1	1
Extra Trees Regressor	n_estimators	100	200
	max_depth	None	None
	min_samples_split	2	2
	min_samples_leaf	1	1

was an increase in the performance of all the algorithms. Support Vector Regressor showed the worst performance, while decision trees-based algorithms showed an R^2 relatively close to Linear Regression, which in turn showed an R^2 of 0.87, making it the best algorithm for this stage. Figure 1 shows a scatter plot relating the shear stress values obtained experimentally and the values predicted by the model.

Table 10 shows the results for the diametral compression tests. The increase in performance of the algorithms was smaller when compared to the direct shear

Table 9 Test phase performances after grid search for direct shear tests

Algorithm	R^2
Linear Regression	0.87
Support Vector Regressor	0.65
Decision Tree Regressor	0.78
Random Forest Regressor	0.80
Extra Trees Regressor	0.84

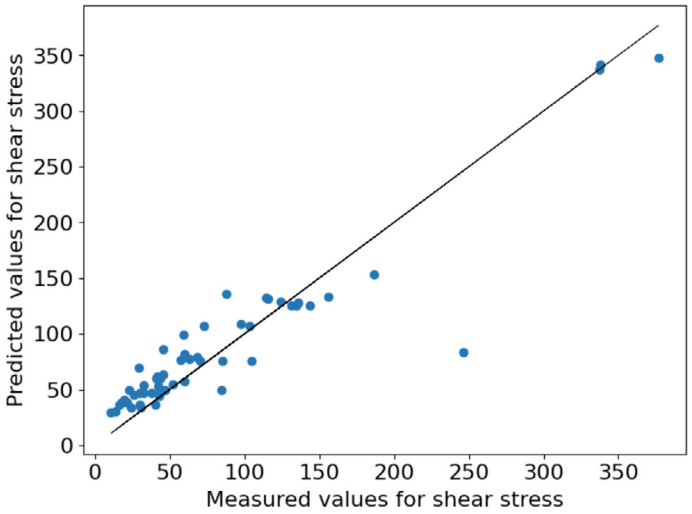


Fig. 1 Relation between experimental and predicted values for shear stress

test. Support Vector Regressor still showed unsatisfactory results, with an R2 of -0.27. The decision tree-based algorithms showed very similar metrics, with 0.79 for Random Forest Regressor and 0.80 for both Decision Tree Regressor and Extra Trees Regressor, the latter being the algorithm chosen as the most suitable for estimating tensile strength. Figure 2 shows a scatter plot relating the tensile strength values obtained experimentally and the values predicted by the model.

Table 10 Test phase performances after grid search for Brazilian tests

Algorithm	R^2
Linear Regression	0.28
Support Vector Regressor	-0.27
Decision Tree Regressor	0.80
Random Forest Regressor	0.79
Extra Trees Regressor	0.80

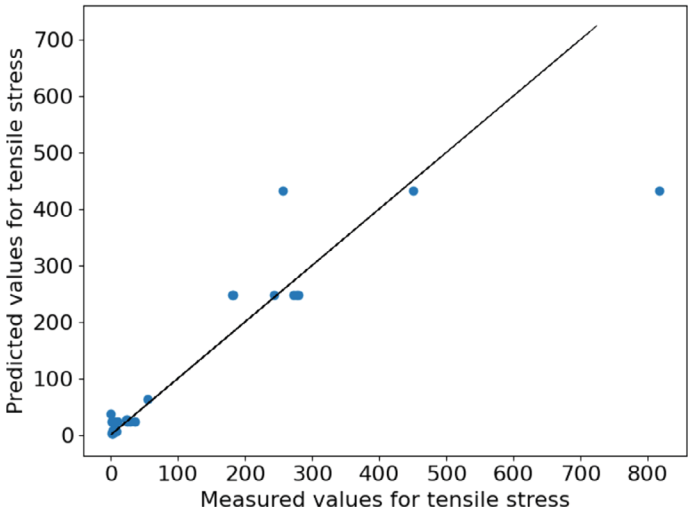


Fig. 2 Relation between experimental and predicted values for tensile stress

5 Conclusions

Overall, the algorithms displayed a satisfactory performance, both with the scikit-learn standard hyperparameter values and after optimization. Although algorithms of several levels of complexity were used, Linear Regression presented the highest R^2 for direct shear tests. It is believed that this is due to the feature "Normal stress (kPa)" which, as it can be seen in Table 3, not only has a high correlation value with the dependent variable, but also has a much higher value than the other variables. In addition, normal stress describes a test parameter, as described in Sect. 2, which directly influences the shear stress obtained, as can be seen in Fig. 1.

For the estimation of Brazilian test results, although the Decision Tree Regressor and the Extra Trees Regressor algorithms showed the same performance, Extra Trees Regressor with optimized hyperparameters specified in Table 8 is defined as the most suitable model. This choice was based on the structure of the algorithm, which ensembles decision trees with the aim of reducing model bias.

This study demonstrates that machine learning algorithms are able to capture complex patterns in soil qualities and to predict strength envelopes with a certain degree of accuracy. It is important to emphasize the ongoing need to improve these models, with applications to other algorithms and using other databases. With continued development in this field, it is hoped that the application of machine learning to geotechnics becomes an increasingly relevant tool for optimizing the process of solving civil engineering problems.

References

1. Vargas, M.: Introduction to Soil Mechanics (in Portuguese). McGraw Hill Brasil, São Paulo (1978)
2. Pinto, C. S.: Basic Course in Soil Mechanics (in Portuguese). Oficina de Textos, São Paulo (2006)
3. Rodriguez, T. T.: Geotechnical Classification Proposal for Brazilian Colluvium (in Portuguese). Doctoral Thesis, Alberto Luiz Coimbra Institute of Postgraduate Studies and Research in Engineering, Rio de Janeiro (2005)
4. Albuquerque, E. A. C., Borges, L. P. F., Cavalcante, A. L. B., Machado, S. L.: Prediction of soil water retention curve based on physical characterization parameters using machine learning. Soils and Rocks. 45(3) (2022). https://doi.org/10.28927/SR.2022.000222
5. Carvalho, L. O., Ribeiro, D. B.: Soil classification system from cone penetration test data applying distance-based machine learning algorithms. Soils Rocks. 42(2), 167–178 (2019). https://doi.org/10.28927/SR.422167
6. Carvalho, L. O., Ribeiro, D. B.: Application of kernel k-means and kernel x-means clustering to obtain soil classes from cone penetration test data. Soils Rocks. 43(4), 607–618 (2020). https://doi.org/10.28927/SR.434607
7. Gerscovich, D.: Slopes Stability (in Portuguese). Oficina de Textos, São Paulo (2012)
8. Fabre, J.S.: Influence of tensile strength on the determination of soil strength parameters (in Portuguese). Federal University of Juiz de Fora, Juiz de Fora, Graduation project (2016)
9. Das, B.M., Sobhan, K.: Principles of Geotechnical Engineering. Cengage Learning, Boston (2017)
10. Lacerda, W. A.: A suggested shallow slide mechanism of accidents in the região serrana of the state of rio de janeiro, In: Lacerda, W.A., Palmeira, E. M., Netto, A. L. C., Ehrlich, M. (eds.) Extreme Rainfall Induced Landslides: an International Perspective, pp. 128–140. Oficina de Textos, São Paulo (2014)
11. Carneiro, F.L.L.B., Barcellos, A.: Tensile strength of concrete (in French). Bulletin Rilem **1**(13), 97–108 (1953)
12. Gerón, A.: Hands-on machine learning with Scikit-Learn, Keras, and Tensorflow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Sebastopol (2019)
13. Botella, R., Lo Presti, D., Vasconcelos, K. et al.: Machine learning techniques to estimate the degree of binder activity of reclaimed asphalt pavement. Mater. Struct. 55(112) (2022). https://doi.org/10.1617/s11527-022-01933-9
14. Gadelha, A. A. B.: Estimation of geotechnical parameters of municipal solid waste using machine learning algorithms (in Portuguese). Master's thesis, Federal University of Ceará, Fortaleza (2021)
15. Theobald, O.: Machine learning for Absolute Beginners. Scatterplot Press (2017)
16. Cortes, C., Vapnik, V.: Support-Vector Networks. Mach. Learn. **20**, 273–297 (1995). https://doi.org/10.1007/BF00994018
17. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Taylor & Francis, New York (1984)
18. Breiman, L.: Random Forests. Mach. Learn. **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324
19. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**, 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1
20. La-Côrte, G.H.A.: Influence of the water table level on slope stability (in Portuguese). Federal University of Juiz de Fora, Juiz de Fora, Graduation project (2015)
21. Silva, I. N., Rodriguez, T. T., Souza, J. H., Braz, J. A., Penasso, S. M.: Slope stability analysis in residual soil (in Portuguese). In: VI Conference on Civil Engineering, pp. 169–183. Juiz de Fora (2019)
22. Rosa, J.P.: Slope stability analysis based on strength parameters in the city of Mar de Espanha, MG (in Portuguese). Federal University of Juiz de Fora, Juiz de Fora, Graduation project (2017)

23. Braz, J. A., Carmo, K. O., Silva, I. N., Rodriguez, T. T., Souza, J. H.: Cut slope in an urban high-risk area (in Portuguese). In: XIX Brazilian Conference on Soil Mechanics and Geotechnical Engineering, Vol III, pp. 676–682. Salvador (2018)
24. Oliveira Júnior, M.R.: Slope Stability Analysis Using Probabilistic Methods (in Portuguese). Federal University of Juiz de Fora, Juiz de Fora, Graduation project (2016)
25. Souza, T.F.: Slope stability analysis of BR-116 in the urban section of Leopoldina (in Portuguese). Federal University of Juiz de Fora, Juiz de Fora, Graduation project (2016)
26. Cardoso, I. M., Silva, F. F.. Rodriguez, T. T.: Influence of tensile strength on stability analysis of vertical slopes (in Portuguese). In: XIX Brazilian Conference on Soil Mechanics and Geotechnical Engineering, Vol. III, pp. 659–668. Salvador (2018)
27. Braz, J. A., Rodriguez, T. T., Penasso, S. M., Jeronymo, L. L. The influence of infiltration in unsaturated soil on slope stability (in Portuguese). In: XX Brazilian Conference on Soil Mechanics and Geotechnical Engineering. Campinas (2022)
28. Oliveira, N.A.C.: Study on the Influence of Soil Tensile Strength on Erodibility (in Portuguese). Federal University of Juiz de Fora, Juiz de Fora, Graduation project (2018)
29. Barra, M.F.W.: Influence of soil tensile strength on unsaturated slope stability analysis (in Portuguese). Federal University of Juiz de Fora, Juiz de Fora, Graduation project (2017)
30. Martins, M. P., Rodrigues, M. H. R., Silva, T. O., Pitanga, H. N., Rodrigues, K. H. P.: Influence of loading rate on unconfined compression and splitting tensile strength of compacted soils (in Portuguese). In: XX Brazilian Conference on Soil Mechanics and Geotechnical Engineering. Campinas (2022)
31. Dantas, G. H. S.: Study on the application of the superpave gyratory compactor in clayey soil compaction (in Portuguese). Master's thesis, São Carlos School of Engineering, São Carlos (2013)
32. Rodrigues, K. H. P., Martins, J. V. F., Rodrigues, M. H., Silva, T. O., Pitanga, H. N.: Influence of specimen dimensions on mechanical properties of compacted soils (in Portuguese). In: XX Brazilian Conference on Soil Mechanics and Geotechnical Engineering. Campinas (2022)
33. Moraes Silva, T. R., Nunes, A. L. L. S., Lacerda, W. A.: Shear strength of two lateritic colluvial soils using tests: diametral compression and direct shear (in Portuguese). In: IV Brazilian Conference on Slope Stability, Vol. 1, pp. 193–204. Salvador (2005)
34. Araujo, R.C., Campos, T.M.P.: Use of cone penetration, disintegration, suction, and tensile strength tests to assess erodibility (in Portuguese). Geotecnia. **128**, 67–85 (2013)

Assessment of Unconfined Compressive Strength of Stabilized Soil Using Artificial Intelligence Tools: A Scientometrics Review

Billal Sari-Ahmed, Mohamed Ghrici, Ali Benzaamia, and Jitendra Khatti

Abstract The Unconfined Compressive Strength (UCS) is a significant geotechnical parameter that characterizes the behaviour of soils. However, the conventional method of determining the UCS is often expensive and time-consuming. Also, the determination of UCS by conventional methods is less accurate and reliable because of the maintenance and calibration of instruments. Therefore, many empirical and advanced computational methods have been introduced and established to compute the UCS, and artificial intelligence (AI) is one of them. This study explores AI-based models' efficacy in predicting UCS for stabilized soils, enhanced by additives or waste materials. Research indicates AI tools reliably predict UCS for stabilized soil, with database quality and quantity influencing prediction accuracy. Still, hybrid approaches outperform conventional, machine, advanced machine, and deep learning methods. This review article will also help PhD researchers develop new research ideas using artificial intelligence in geotechnical engineering.

Keywords Artificial intelligence · Geotechnical engineering · Unconfined compressive strength · Hybrid learning · Stabilized soil

1 Introduction

Please note that the AISC Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

B. Sari-Ahmed · M. Ghrici (✉) · A. Benzaamia
Geomaterials Laboratory, University Hassiba Benbouali of Chef, Ouled Fares, Algeria
e-mail: m.ghrici@univ-chlef.dz

J. Khatti
Department of Civil Engineering, Rajasthan Technical University, Kota, Rajasthan, India

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_15

Unconfined Compressive Strength (UCS) of stabilized soils is a valuable parameter in many engineering projects, particularly in the construction and civil engineering sectors [1, 2]. UCS defines the ability of stabilized soil to withstand compressive stresses, which is crucial for ensuring the stability and durability of structures built on these soils [3, 4]. Soil stabilization is a common practice to improve the mechanical properties of natural soils by adding stabilizing materials such as cement, lime, various cementitious additives, or geopolymers. These stabilizing agents reinforce soil and make it suitable for the desired development [5–7]. However, accurately predicting the UCS of stabilized soils remains a major challenge for engineers and researchers. It becomes more crucial during infrastructure design because it directly influences the design and construction decisions and the long-term durability of structures [8, 9].

Conventional methods for predicting the UCS of stabilized soils relied on empirical or semi-empirical models, often based on laboratory tests and experimental data. Although these methods have been widely used, they have limitations in accuracy and generalization, mainly due to the variability of soils and stabilizing materials [10, 11]. In this context, artificial intelligence (AI) techniques are emerging as a promising approach to improve the prediction of UCS for stabilized soils [12, 13]. AI approaches, such as Gaussian process regression (GPR), relevance vector machine (RVM), decision tree (DT), random forest (RF), extreme learning machine (ELM), artificial neural networks (ANN), support vector machines (SVM), etc. offer the ability to process complex datasets and discover nonlinear patterns that may be difficult to detect with traditional approaches [14, 15]. Many recent studies have explored the application of AI techniques in predicting the UCS of stabilized soils, yielding promising results. These approaches often leverage a combination of geotechnical data, characteristics of stabilizing materials, and environmental conditions to develop more accurate and robust prediction models [16–20].

2 Methodology

This review article illustrates the advancement of artificial intelligence techniques in geotechnical engineering to predict the unconfined compressive strength of stabilized soil. For that purpose, several published investigations have been studied and analyzed. These investigations are classified based on the (i) soil stabilizer and the stabilization mechanism, (ii) Applications of AI for stabilized soils, and (iii) limitations of the AI approaches. The section “Soil stabilizers and the stabilization mechanism” categories and discusses the different stabilizing additives. However, many materials, i.e., industrial products and by-products, are being used to improve the strength of the soil. Therefore, this review article discusses the use of AI approaches for cement, mineral additives, and geopolymer-stabilized soils. Conversely, the section “Applications of AI for stabilized soil” discusses the published investigation on the prediction of UCS of stabilized soil. Based on the overall analysis, the

limitations of the AI approaches have been discussed. Finally, the overall study has drawn the conclusions and suggestions for further research.

3 Stabilization Materials and Mechanism

Soil stabilization, a cornerstone of many construction projects, relies on the strategic use of additives to enhance the mechanical properties of expansive soil, including UCS. These diverse additives, with distinct mechanisms of action, are grouped into several categories [7]. Cement and lime are the conventional additives among the pioneers in this field [21, 22]. The cement reacts with soil particles to form chemical bonds, consolidating the structure and increasing strength [23]. On the other hand, lime interacts with clays, reducing their plasticity and improving overall cohesion [24].

Subsequently, industrial by-products offer a sustainable and effective alternative. Fly ash, produced by coal combustion [25], reacts with soil particles to strengthen the matrix, while phosphogypsum, a residue from the phosphoric acid industry, helps reduce plasticity [26]. By-products such as cement kiln, silica fume (SF), and blast furnace slag (BFS) leverage their specific compositions to interact with soils and improve their strength [27–29]. Figure 1 illustrates the different stabilization additives to improve soil strength parameters.

Agricultural by-products, such as eggshell powder (ESP) and rice husk ash (RHA), contribute well to soil treatment [30, 31]. ESP's calcium richness promotes the formation of chemical bonds, while RHA, rich in silica and alumina, strengthens soils and enhances their resistance [30, 31]. Polymers, particularly geopolymers, are an innovative approach to soil treatment. These inorganic polymers form solid matrices, enhancing the soil's resistance to compression [32]. These different additives act

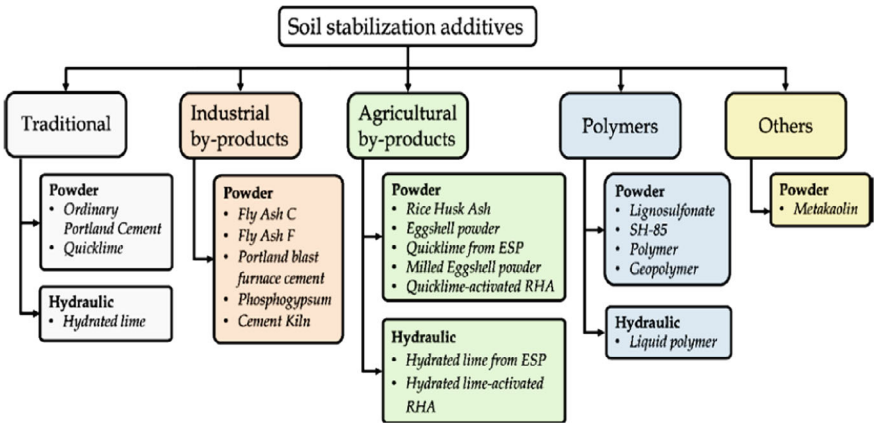


Fig. 1 Categorization of additives employed in soil stabilization [7]

through various mechanisms: formation of chemical bonds, hydration of components, reduction of plasticity, and improvement of soil compaction. By judiciously combining these additives and tailoring them to the specific needs of each project, it is possible to obtain stabilized soils with optimized mechanical properties, ready to support even the most demanding structural loads.

4 Application of Artificial Intelligence for Stabilized Soils

The integration of artificial intelligence (AI) techniques in the field of geotechnical engineering has significantly evolved, offering a promising assessment of unconfined compressive strength (UCS) of stabilized soils [8, 33, 34]. AI is the most potent tool and solves geotechnical engineering issues using large databases [9, 14, 35]. Artificial intelligence techniques give the most promising results of UCS of stabilized soil compared to conventional methods, i.e., simple and multilinear regression [9, 10]. An infrastructure becomes safe and sustainable if an accurate UCS is predicted [1]. The prediction accuracy is based on the type of artificial intelligence technique used. Still, the relationship among various variables cannot be ignored when assessing the UCS of stabilized soil, i.e., curing period, additive percentage, consistency limits, and compaction parameters [1, 36, 37]. The artificial neural network, genetic algorithm, and support vector machine are the most common AI techniques used to assess the UCS of stabilized soil [1, 12, 13, 17, 38]. Many artificial intelligence methods are available to predict the UCS of the stabilized soil, as shown in Fig. 2, but they are not limited [38].

Each technique has specific advantages, such as the ability of neural networks to model complex nonlinear relationships and the robustness of SVMs against small datasets [9, 39]. The AI methods have the following limitations: (i) sensitivity to noisy data, (ii) the need for large amounts of training data, and (iii) the complexity of model interpretation. Several approaches are being explored to overcome these limitations. One strategy involves enhancing data quality by meticulously cleaning and preprocessing datasets to reduce noise. Additionally, regularization techniques and data augmentation can be applied to optimize model performance with smaller datasets [40]. The following subsections discuss the different investigations using AI methods to predict the UCS of cement, minerals (lime, fly ash, slag, etc.), and geopolymer-stabilized soils.

4.1 Cement-Stabilized Soils

Soil stabilization with cement aims to enhance engineering properties like workability and shear strength by integrating cement into the soil matrix [23]. Cement boosts hydration, fills soil voids, and produces chemicals that enhance soil strength [22]. The quantity of cement significantly affects compressive strength, with higher

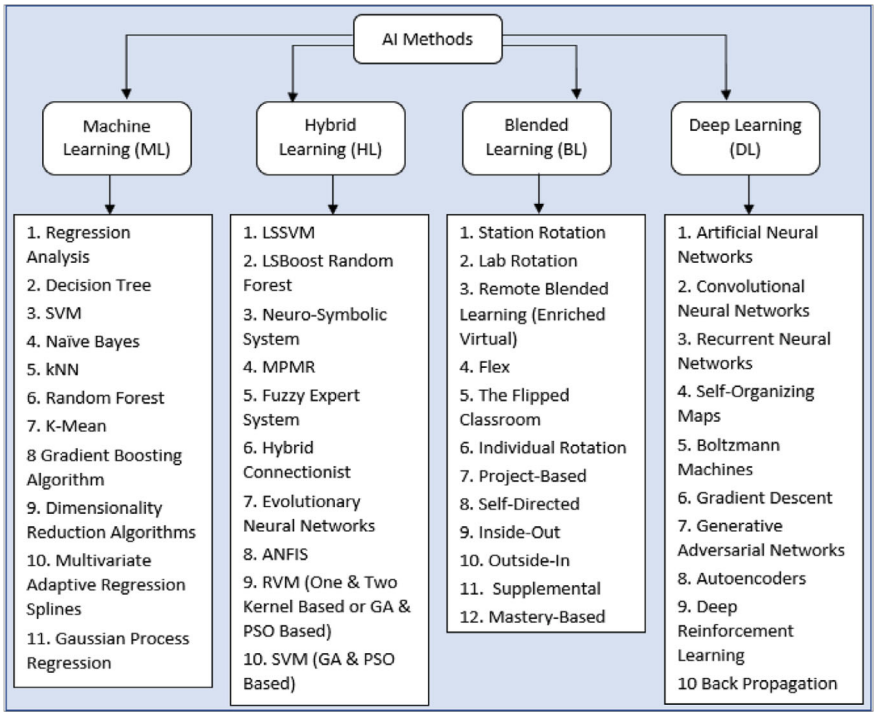


Fig. 2 Soft Computing Techniques in Artificial Intelligence [38]

amounts leading to increased strength over time [22]. Soil particle size distribution also affects stabilization, with finer particles promoting better bonding. Both field and lab studies demonstrate a significant improvement in the compressive strength of soil, validating its effectiveness [23].

AI techniques in assessing the UCS of cement-stabilized soils have been increasingly explored in recent years. Studies utilized machine learning algorithms such as artificial neural networks ANN and support vector machines (SVM) to estimate the UCS of stabilized soils [2, 41]. These approaches revealed that the cement content, moisture content, and soil characteristics play an essential role in predicting the UCS of cement-stabilized soil [42, 43]. Additionally, hybrid evolutionary models combining optimization algorithms with ANN were developed to improve the prediction capabilities for UCS of cement-stabilized soils [8]. These advanced computational frameworks obtained promising results in capturing the nonlinear relationships between UCS and its influencing factors, providing valuable tools for engineers in soil mechanics and geotechnical engineering applications [8]. Table 1 summarizes the published research for predicting the UCS of cement-stabilized soil.

Table 1 reveals that a huge utilization of artificial neural networks [34, 41, 46, 47], genetic algorithms [8], evolutionary polynomial regressions [45], and support vector machines [2, 41, 47] in predicting the UCS of cement-stabilized soil. This

Table 1 Summary of published models for predicting the UCS of cement-stabilized soil

No	References	Year	Soft computing technique	Soil Type	Data size	Input variables	Best Performance
1	[44]	2024	LM-BP-ANN, SCG-BP-ANN	RAP	64	FR, CR, NAC, NAF, CS, BR, OMC, CC	R = 0.9900
2	[2]	2024	SVR, DTR	Organic soils	450	Si-I, Fe-I, Mg-I, Ca-I, Na-I, Al-I, Zn-II, In-II, Ti-I, O-I and K-I, Bd, WC	R = 0.9949
3	[34]	2023	ANN	Clayey soil (CH, CL, MH)	192	CC, FC, S, G, t, PI, W/C, UCS_0	R = 0.9899
4	[45]	2022	EPR	Clay (CL)	75	CC, WC, t	R = 0.9798
5	[8]	2021	GA-ANN, PSO-ANN, ICA-ANN	Unsaturated soil	96	CC, γ_d, Suc	R = 0.9899

(continued)

Table 1 (continued)

No	References	Year	Soft computing technique	Soil Type	Data size	Input variables	Best Performance
6	[46]	2021	ANN	Sandy soil	80	F0.5, F0.25, F0.1, SiO_2, Fe_2O_3, Al_2O_3, SO_3, K_2O, CaO, Ti_2O, CC, t	R = 0.9949
7	[47]	2019	SVM, ANN, MR	Cohesive and Organic soils	444	CF, SC, Silt, OM, WC, CC, W/C and t	R = 0.9747
8	[43]	2016	MARS, FN	Soft soil	51	LL, PL, SC, GC, MC, CC	R = 0.9747
9	[41]	2011	ANN, SVM	Undefined soil classes	55	LL, PI, CF, GC, MC, CC	R = 0.9539
10	[42]	2006	MLP, RBF, GP	Clayey soils (CH, CL)	47	t, W/C, CC, WC, pH, Na^+, LL, LI	R = 0.9949

LM-BP-ANN is the levenberg–marquardt backpropagation-based ANN, SCG-BP-ANN is the scaled conjugate gradient backpropagation-based ANN, RAPM is the reclaimed asphalt pavement material, FR is the fine RAPM, CR is the coarse RAPM, NAC is the natural aggregate coarse, NAF is the natural aggregate fine, CS is the chemical stabilizer, BR is the recovered bitumen, OMC is the optimum moisture content, CC is the cement content, R is the correlation coefficient, ANN is the artificial neural network, CH is the inorganic clay of high plasticity, CL is the inorganic clay of low plasticity, MH is the silts of high plasticity, FC is the fine content, SC is the sand content, GC is the gravel content, t is the curing time, PI is the plasticity index, W/C is the water-cement ratio, UCS0 is the unconfined compressive strength of virgin soils, SVR is the support vector regression, DTR is the decision tree regression, Si-I, Fe-I, Mg-I, Ca-I, Na-I, Al-I, Zn-II, In-II, Ti-I, O-I and K-I are the levels of radiation emitted by cement-stabilized soil components during laser-induced spectroscopy (LIBS), Bd is the bulk density, WC is the Water content, EPR is the evolutionary polynomial regression, R2 is the coefficient of determination, GA-ANN, PSO-ANN and ICA-ANN are the genetic algorithm, particle swarm optimization and imperialist competitive algorithm, hybridized with artificial neural network, γd is the Dry density, Suc is the Suction, F0.5, F0.25 and F0.1 are the percentages of the soil particles passed through the sieves 0.5, 0.25 and 0.1 mm, respectively, SiO2, Fe2O3, Al2O3, SO3, K2O, CaO, Ti2O are the chemical composition of the soils, SVM is the support vector machine, MR is the multiple regression, CF is the clay content, Silt is the Silt content, OM is the organic matter content, MARS is the multivariate adaptive regression splines, FN is the functional networks, LL is the liquid limit, PL is the plastic limit, PI is the plasticity index, MC is the moisture content MLP is the multilayer perceptron, RBF is the radial basis function, pH is the potential of hydrogen, Na + is the sodium ion concentration, LI is the Liquidity index.

variety attests to the flexibility of AI approaches in this field. Regarding the types of improved soils, an extensive range is noted, including clayey, organic, and sandy soils [42, 45–47]. The input parameters used in these models encompass various physicochemical soil characteristics including texture, organic matter content, pH, and bulk density [42, 45, 47], geotechnical properties such as soil classification, particle size distribution, Atterberg limits, and compaction properties [34, 43], as well as treatment variables like cement content and curing time [34, 45]. These parameters provide a comprehensive view of the factors influencing the compressive strength of cement-stabilized soils. The performance of the models was measured in terms of coefficient of determination, correlation coefficient, and mean squared error [8].

Narendra et al. [42] investigated clayey soils employing multiple AI techniques such as multilayer perceptron (MLP), radial basis function (RBF), and genetic programming (GP). The investigators specifically focused on parameters like curing time, water-cement ratio, and chemical composition, achieving an R^2 value of 0.99 [42]. Similarly, Tinoco et al. [47] utilized support vector machine (SVM), artificial neural networks (ANN), and multiple regression (MR) on cohesive and organic soils. The authors considered clay content, water content, and curing time as input parameters [47]. Moreover, Mojtahedi et al. [34] employed an ANN model using clay content, liquid limit, plasticity index, and water-cement ratio parameters, achieving an R^2 of 0.98 [34]. The analysis of the published investigation reveals a positive sign for implementing artificial intelligence techniques to predict the cement-stabilized soil with high accuracy and less prediction error.

4.2 Mineral Stabilized Soil

Soil stabilization through mineral additives, particularly natural ones like lime and Zeolite, alongside industrial by-products (such as Fly ash (FA), pond ash (PA), cement kiln dust (CKD), silica fume (SF), asphalt), and agricultural by-products like eggshell powder (ESP) and rice husk ash (RHA), has garnered significant attention in research [25–27, 29–31, 48, 49]. These additives are instrumental in bolstering stabilized soils' UCS. The soil type, additive composition, and curing conditions significantly enhance the UCS [7]. Artificial intelligence (AI) techniques have been employed to predict the UCS of mineral-stabilized soil, as summarized in Table 2 [9, 11–16, 35, 50–55]

Table 2 demonstrates that most researchers utilized the machine and deep learning approaches to predict the UCS of mineral-treated soils. It can be observed that the artificial intelligence tool assessed the UCS with a performance of more than 0.95, which was acceptable and close to the test results. Various studies explored those additives such as fly ash, lime, cement, pond ash, rice husk ash, and asphalt, either individually or combined, improve soil properties across different soil types, including black-cotton soil, cohesive soil, clayey soils, organic soils, sulfate silty sand, and poorly graded sand. For computing the UCS, fly ash content, lime content, cement

Table 2 Summary of the published investigation on mineral-stabilized soil

No	References	Year	Soft computing technique	Soil Type	Data size	Precursor material	Input variables
1	[55]	2024	ANN	Black-cotton soil	125	FA, lime cement	FAc, LC, CC, LL, PL, MDD, OMC
2	[54]	2024	GPR, ET, DT, SVM, RVM	Cohesive soil	371	Lime	UCS_0, LC, t
3	[13]	2024	ANN	Clayey soils	1120	Lime	LL, PL, γ_d, WC, FC, T, LC, t
4	[11]	2024	RF, CART, MLR	Clayey soil (CI)	45	FA and cement	CC, FAc, t
5	[15]	2023	SVR-SA, SVR-RRHC, SVR-PSO, SVR-HGS, SVR-SMA	Organic soils	227	Lime and cement	Scenario 1: OM, SC, Silt, CF, GC, LL, PL, PI, LC, CC
6	[9]	2023	ANN	Clayey soil	187	Lime and cement	S, CC, LC, LL, PL, PI, MDD, OMC
7	[53]	2022	MEP	kaolin and BC soils (CL, CH)	384	FA	FAc, ALKC, t
8	[12]	2020	ANN	Clay (CL)	72	CKD	SG, LS, Cu, Cc, LL, PL, OMC, MDD

(continued)

Table 2 (continued)

No	References	Year	Soft computing technique	Soil Type	Data size	Precursor material	Input variables
9	[16]	2020	ANN	Clayey soil (CL)	129	PA, RHA, cement	CF, PC, RC, CC, t
10	[35]	2019	LR, DA, KNN, SVM	Clayey soil	193	cement, lime, asphalt	LL, PL, FC, SC, LC, CC, ASPH
11	[52]	2018	RBF-NN, EPR	Sulfate silty sand	90	MS and lime	LC, MSP, t, Cd, CBR_{10}
12	[14]	2016	GMDH	Poorly graded sand	216	Zeolite and cement	CC, ZC, Dr, t
13	[51]	2014	GEP	Clay (CL)	70	BA	BAc, γd, Cr, I_B b, Ed
14	[50]	2009	RBF-NN	Clayey soil	230	Cement, lime, asphalt	LL, PL, PI, LS, GC, CF/Silt, SC, LC, CC, ASPH

ANN is the artificial neural network, *FA* is the Fly ash, *FAc* is the fly ash content, *LC* is the lime content, *CC* is the cement content, *LL* is the liquid limit, *PL* is the plastic limit, *PI* is the *plasticity index,* R^2 is the coefficient of determination, *GPR* is the Gaussian process regression, *ET* is the ensemble tree, *DT* is the decision tree, *SVM* is the support vector machine, *RVM* is the relevance vector machine, UCS_0 is the unconfined compressive strength of virgin soils, *t* is the curing time, *R* is the correlation coefficient, γ_d is the Dry density, *WC* is the Water content, *FC* is the Fine content, *T* is the Temperature, *RF* is the random forest, *CART* is the classification and regression trees, *MLR* is the multiple regression analysis, *CI* is the inorganic clay of medium plasticity, *SVR-SA, SVR-RRHC, SVR-PSO, SVR-HGS and SVR-SMA* are the support vector regression integrated with metaheuristic algorithms such as simulated annealing, random restart hill climbing, particle swarm optimization, hunger games search, and slime mould algorithm, *OM* is the organic matter content, *SC* is the sand content, *CF* is the clay content, *GC* is the gravel content, *S* is the soil percentage, PI is the plasticity index, *MDD* is the maximum dry density, *OMC* is the optimum moisture content, *MEP* is the multi expression programming, *CL* is the inorganic clays of low plasticity, *CH* is the inorganic clays of high plasticity, *ALKC* is the Alkali Concentration, *CKD* is the ciment kiln dust, *SG* is the specific gravity, *LS* is the linear shrinkage, *Cu* is the uniformity coefficient, *Cc* is the coefficient of gradation, *SG* is the specific gravity, *LS* is the linear shrinkage, *Cu* is the uniformity coefficient, *Cc* is the coefficient of gradation, *PA* is the Pond Ash, *RHA* is the Rice Husk Ash, *PC* is the pond ash content, *RC* is the rice husk ash content, *LR* is the logistic regression, *DA* is the discriminant analysis, *KNN* is the k-nearest neighbors, *SVM* is the support vector machines, *ASPH* is the asphalt content, *TPR* is the true positive rate, *CPR* is the correct prediction rate, *RBF-NN* is the radial basis function neural network, *EPR* is the evolutionary polynomial regression, MS is the *Micro Silica*, *MSP* is the micro silica content, *Cd* is the curing conditions, CBR_{10} is the California bearing ratio of samples compacted with 10 blow counts, *GMDH* is the group method of data handling, *ZC* is the Zeolite content, *Dr* is the relative density, *GEP* is the enetic expression programming, *BA* is the bottom ash, *BAc* is the bottom ash dosage, γd is the dry unit weight, *Cr* is the relative compaction, I_B is the brittleness index, *Ed* is the energy absorption capacity, *RBF-NN* is the radial basis function neural network, *LS* is the linear shrinkage, *CF/Silt* is the clay-silt ratio.

content, liquid limit, plastic limit, plasticity index, maximum dry density, optimum moisture content, curing time, and various soil properties were used. For instance, Güllü [51] employed Genetic Expression Programming (GEP) to predict the UCS of clay soils stabilized with bottom ash (BA). Priyadarshee et al. [16] achieved the high accuracy of the ANN model in predicting the UCS of clayey soils stabilized with pond ash (PA), rice husk ash (RHA), and cement. Khan et al. [53] obtained robust predictive capabilities with Multi Expression Programming (MEP) for kaolin and BC soils stabilized with fly ash (FA). Taleb-Bahmed et al. [54] achieved notable results in UCS prediction for cohesive soils stabilized with lime using various techniques, including Gaussian process regression (GPR), ensemble tree (ET), support vector machine (SVM), and hybrid relevance vector machine (RVM). In conclusion, the amalgamation of mineral additives for soil stabilization, alongside the application of artificial intelligence techniques for UCS prediction, presents a promising avenue for advancing soil engineering practices. These studies highlight the intricate relationship between additive composition, soil properties, and predictive parameters in achieving effective soil stabilization outcomes.

4.3 Geopolymer Stabilized Soil

Soil stabilization by geopolymers represents a significant advancement in construction and civil engineering, offering notable environmental benefits and cost-effectiveness [32, 56]. Geopolymers are binding materials obtained through the polymerization of precursors rich in alumina and silicate, such as fly ash, slag, or metakaolin, in the presence of alkaline activators, which is generally a solution of sodium hydroxide or potassium hydroxide [56, 57]. This chemical reaction forms a solid three-dimensional network, imparting to the treated soils improved mechanical strength, decreased permeability, and increased stability against cycles of wetting and drying [32, 56, 58, 59]. The impact of geopolymers on the unconfined compressive strength (UCS) of stabilized soils has been extensively studied [37, 57, 60]. The results showed a significant improvement in the UCS of soils after treatment with geopolymers [37, 57–59]. However, precisely predicting this strength remains challenging due to the complex interactions between materials and environmental conditions [1, 59]. To address this challenge, artificial intelligence (AI) techniques offer promising solutions [36, 57, 61]. In this context, numerous researchers have proposed AI-based methodologies to predict the UCS of geopolymer-stabilized soil, employing soft computing techniques, such as Gaussian Process Regression (GPR), Support Vector Machine (SVM), Random Forest (RF), Adaptive Neuro-Fuzzy Inference System (ANFIS), Artificial Neural Network (ANN), Group Method of Data Handling (GMDH), and Multi-Genetic Programming (MGGP) [1, 36, 57, 62, 63]. Table 3 summarizes some of these studies.

Table 3 shows that each author used a different input parameter. Still, the physicochemical characteristics of stabilizing materials, such as the content of hydrated lime-activated rice husk ash (HARHA), granulated blast furnace slag (GBFS), as

Table 3 Summary of the published models in predicting geopolymer-stabilized soil

No	References	Year	Soft computing technique	Soil Type	Data size	Precursor material	Input variables
1	[1]	2023	GPR, SVM	Clayey soil	121	Rice husk ash	HARHA, LL, PL, PI, OMC, AC, MDD
2	[57]	2023	RF, BRRF	clayey soil (CH, CL)	286	FA, GBFS	LL, IP, GBFSc, FAc, M, A/B, Na/Al, Si/Al
3	[61]	2023	RF	clayey soil (CH, CL)	283	FA, GBFS	PI, GBFSc, FAc, M, A/B, Na/Al, Si/Al
4	[36]	2021	ERF, ANFIS	clayey soil	61	Rice husk ash	HARHA, LL, LP, IP, OMC, AC, MDD
5	[63]	2020	ANN and GMDH-NN	clayey soil (CH, CL)	213	GBFS	PI, GBFSc, M, A/B, Na/Al, Si/Al
6	[64]	2019	GMDH	Cohesive soils	283	FA, GBFS	PI, LL, GBFSc, FAc, M, A/B, Na/Al, Si/Al
7	[63]	2018	MGGP	Clayey soil	283	FA, GBFS	FAc, GBFSc, LL, PL, PI, M, A/B, Na/Al and Si/Al
8	[65]	2017	SVR	Clayey soil (CH, CL)	283	GGBS, GGFS + FA	GBFSc, LL, PI, M, A/B, Na/Al, Si/Al
9	[37]	2015	ANN	Clayey soil (CH, CL)	283	GGFS, GGFS + FA	GBFSc, FAc, LL, PI, M, A/B, Na/Al, Si/Al

GPR is the Gaussian process regression, SVM is the support vector machine, HARHA is the hydrated-lime-activated rice husk ash, LL is the liquid limit, PL is the plastic limit, PI is the plasticity index, OMC is the optimum moisture content, AC is the clay activity, MDD is the maximum dry density, R^2 is the coefficient of determination, RF is the random forest, BRRF is the random forest with random search hyperparameter, CH is the inorganic clays of high plasticity, CL is the inorganic clays of low plasticity, FA is the fly ash, GBFS is the ground granulated blast furnace slag, GBFSc is the ground granulated blast furnace slag content, FAc is the fly ash content, M is the molar concentrations of alkali solution, A/B is the activator to binder ratio, Na/Al is the sodium-to-alumina atomic ratio, Si/Al is the silicate-to-alumina atomic ratio, ERF is the ensemble random forest, ANFIS is the adaptive neuro-fuzzy inference system, R is the correlation coefficient, ANN is the artificial neural network, GMDH-NN is the neural network-based group method of data handling, GMDH is the group method of data handling, MGGP is the multi-gen genetic programming, MAPE is the mean absolute percentage error.

well as soil properties such as liquid limit (LL), plastic limit (PL), plasticity index (PI), optimal moisture content (OMC), maximum dry density (MDD), and others have been considered in the prediction [1, 61].

5 Discussion on Investigation

The findings presented in this investigation underscore the significant potential of artificial intelligence (AI) techniques in predicting the unconfined compressive strength (UCS) of stabilized soils. The ability to accurately model and capture the intricate relationships between soil properties, stabilizing additives, and environmental factors is crucial for optimizing soil stabilization practices and ensuring the long-term performance of constructed infrastructure. Integrating AI models into geotechnical engineering workflows can revolutionize the design and construction processes. By leveraging the predictive capabilities of these models, engineers can make more informed decisions regarding soil treatment strategies, additive dosages, and construction specifications. This data-driven approach can improve soil stabilization projects' efficiency, sustainability, and cost-effectiveness. Furthermore, the ability to rapidly evaluate different scenarios and conduct virtual experiments using AI models can streamline the iterative design process, reducing the need for extensive laboratory testing and field trials. This can lead to significant time and cost savings while minimizing the environmental impact of traditional empirical methods. However, it is observed that the optimized computational models give promising and reliable results in predicting the UCS of stabilized soil.

6 Summary and Conclusions

This article thoroughly examined the application of artificial intelligence techniques in predicting the unconfined compressive strength of stabilized soils, particularly on soils stabilized by cement, mineral additives, and geopolymers. The reviewed research demonstrated the ability of artificial intelligence models, such as artificial neural networks, support vector machines, and hybrid techniques, to accurately predict these soils' unconfined compressive strength (UCS). In the case of soils stabilized by cement, it is clear that the amount of cement, the particle size distribution of soil particles, and other characteristics play a significant role in UCS. Studies showed that artificial intelligence models can handle these complex relationships and provide valuable tools for civil engineering professionals. Similarly, research on soils stabilized by mineral additives such as lime, fly ash, and blast furnace slag has revealed the diversity of artificial intelligence approaches used to predict UCS. These studies highlight the importance of additive content, soil properties, and curing conditions in accurately predicting UCS.

Moreover, geopolymers offer an innovative solution for soil stabilization, and artificial intelligence models have been effectively applied to predict their compression resistance. Research has shown that the composition of geopolymer precursors and soil properties are key factors in predicting UCS. The findings underscore the promising potential of AI in the field of geotechnical engineering and soil stabilization. By leveraging the predictive capabilities of these models, engineers can make well-informed decisions regarding soil treatment strategies, additive dosages, and construction specifications, leading to efficiency, sustainability, and cost-effectiveness improvements across the entire spectrum of soil stabilization projects.

While current models have shown promising performance, several avenues warrant further exploration to unlock the full potential of AI in this domain. These include expanding the scope of input parameters, exploring ensemble and hybrid modelling techniques, enhancing interpretability and explainability, incorporating uncertainty quantification and sensitivity analysis, and fostering real-world validation and implementation through collaboration between researchers, practitioners, and industry stakeholders.

By continuously advancing AI techniques in geotechnical engineering and addressing the recommendations outlined in this study, the construction industry can significantly enhance its capabilities in delivering sustainable, cost-effective, and high-performance infrastructure projects while contributing to the broader goal of sustainable development and environmental stewardship.

7 Suggestions for Further Research

The following suggestions for further research have been drawn based on the outcomes of the published work and summary:

- The published research may be extended by including and analyzing the effect of environmental factors, i.e., temperature, humidity, and precipitations. Also, the impact of soil mineralogy and chemical composition may be used as input variables.
- Develop, train, test, and analyze the metaheuristic algorithms, i.e., evolutionary, physical, natural, swarm-based, and biological, optimized soft computing models for predicting the UCS of stabilized soil.
- The effect of the training database's quality and quantity on the performance of machine, advanced machine, deep, and hybrid learning models in predicting the UCS of stabilized soil may be analyzed.
- The effect of multicollinearity may be analyzed for the machine, advanced machine, deep, and hybrid learning models in predicting the UCS of stabilized.

These suggestions will help geotechnical engineers and designers develop, train, test, and analyze a robust artificial intelligence tool to predict the UCS of stabilized soil for mega projects of civil engineering.

References

1. Ahmad, M., Al-Mansob, R.A., Ramli A.B.B., Ahmad F., Khan, B.J.: Unconfined compressive strength prediction of stabilized expansive clay soil using machine learning techniques. Multiscale and Multidisciplinary Modeling. Exp. Des. **7**, 217–231 (2024)
2. Wudil, Y.S., Al-Osta, M.A., Al-Amoudi, O.S.B., Gondal, M.A.: Integrating laser-induced breakdown spectroscopy and non-linear random forest-based algorithms to predict soil unconfined compressive strength. Environ. Earth Sci. **83**, 151 (2024)
3. Sari Ahmed, B., Gadouri, H., Ghrici, M., Harichane, K.: Best-fit models for predicting the geotechnical properties of FA–stabilised problematic soils used as materials for earth structures. Int. J. Pavement Eng. **21**(7), 939–953 (2020)
4. Sari-Ahmed, B., Ghrici, M., Harichane, K.: Predicting the UCS of Lime-Stabilized Clayey Soils. Geotech. Eng. **53**(3), 00465828 (2022)
5. Harichane, K., Ghrici, M., Kenai, S.: Effect of the combination of lime and natural pozzolana on the compaction and strength of soft clayey soils: a preliminary study. Environ. Earth Sci. **66**, 2197–2205 (2012)
6. Baldovino, J.A., Moreira, E.B., Teixeira, W., Izzo, R.L., Rose, J.L.: Effects of lime addition on geotechnical properties of sedimentary soil in Curitiba, Brazil. J. Rock Mech. Geotech. Eng. **10**(1), 188–194 (2018)
7. Anburuvel, A.: The engineering behind soil stabilization with additives: a state-of-the-art review. Geotech. Geol. Eng. **42**(1), 1–42 (2024)
8. Kardani, N., Zhou, A., Shen, S.L., Nazem, M.: Estimating unconfined compressive strength of unsaturated cemented soils using alternative evolutionary approaches. Transp. Geotech. **29**, 100591 (2021)
9. Kumar, A., Sinha, S., Saurav, S.: Random forest, CART, and MLR-based predictive model for unconfined compressive strength of cement reinforced clayey soil: A comparative analysis. Asian J Civ Eng. **25**, 2307–2323 (2023)
10. Skentou, A.D., Bardhan, A., Mamou, A., Lemonis, M.E., Kumar, G., Samui, P., Armaghani, D.J., Asteris, P.G.: Closed-form equation for estimating unconfined compressive strength of granite from three non-destructive tests using soft computing models. Rock Mech. Rock Eng. **56**, 487–514 (2023)
11. Kumar, A., Singh, V., Singh, S., Kumar, R., Bano, S.: Prediction of unconfined compressive strength of cement–lime stabilized soil using artificial neural network. Asian J Civ Eng. **25**, 2229–2246 (2024)
12. Salahudeen, A., Sadeeq, J.A., Badamasi, A., Onyelowe, K.C.: Prediction of unconfined compressive strength of treated expansive clay using back-propagation artificial neural networks. Niger. J. Eng. **27** (2020)
13. Muhmed, A., Mohamed, M., Khan, A.: Prediction of unconfined compressive strength of lime treated soils. Geomech. Geoengin. 1–17 (2024)
14. Mola-Abasi, H., Shooshpasha, I.: Prediction of zeolite-cement-sand unconfined compressive strength using polynomial neural network. Eur. Phys. J. Plus. **131**, 108 (2016)
15. Ngo, T.Q., Nguyen, L.Q., Tran, V.Q.: Novel hybrid machine learning models including support vector machine with meta-heuristic algorithms in predicting unconfined compressive strength of organic soils stabilised with cement and lime. Int. J. Pavement Eng. **24**(2), 2136374 (2023)
16. Priyadarshee, A., Chandra, S., Gupta, D., Kumar., V.: Neural Models for Unconfined Compressive Strength of Kaolin clay mixed with pond ash, rice husk ash and cement. J. Soft Comput. Civ. Eng. **4**(2), 85–102 (2020)
17. Onyelowe, K.C., Ebid, A.M., Onyia, M.E., Nwobia, L.I.: Predicting nanocomposite binder improved unsaturated soil UCS using genetic programming. Nanotechnol. Environ. Eng. **6**, 39 (2021)
18. Hoque, M.I., Hasan, M., Islam, M.S., Houda, M., Abdallah, M., Sobuz, M.H.R.: Machine Learning Methods to Predict and Analyse Unconfined Compressive Strength of Stabilised Soft Soil with Polypropylene Columns. Cogent Eng. **10**(1), 2220492 (2023)

19. Li, D., Zhang, X., Kang, Q., Tavakkol, E.: Estimation of unconfined compressive strength of marine clay modified with recycled tiles using hybridized extreme gradient boosting method. Constr. Build. Mater. **393**, 131992 (2023)
20. Talamkhani, S.: Machine Learning-Based Prediction of Unconfined Compressive Strength of Sands Treated by Microbially-Induced Calcite Precipitation (MICP): A Gradient Boosting Approach and Correlation Analysis. Adv. Civ. Eng. 3692090 (2023)
21. Ali, H., Mohamed., M.: The effects of lime content and environmental temperature on the mechanical and hydraulic properties of extremely high plastic clays. Appl. Clay Sci. **161**, 203–210 (2018)
22. Pu, S., Zhu, Z., Song, W., Wan, Y., Wang, H., Song, S., Zhang, J.: Mechanical and microscopic properties of cement stabilized silt. KSCE J. Civ. Eng. **24**, 2333–2344 (2020)
23. Eskisar, T.: Influence of cement treatment on unconfined compressive strength and compressibility of lean clay with medium plasticity. Arab. J. Sci. Eng. **40**, 763–772 (2015)
24. Aldaood, A., Bouasker, M., Al-Mukhtar, M.: Impact of freeze–thaw cycles on mechanical behaviour of lime stabilized gypseous soils. Cold Reg. Sci. Technol. **99**, 38–45 (2014)
25. Sezer, A., İnan, G., Yılmaz, H.R., Ramyar, K.: Utilization of a very high lime fly ash for improvement of Izmir clay. Build. Environ. **41**(2), 150–155 (2006)
26. Degirmenci, N., Okucu, A., Turabi, A.: Application of phosphogypsum in soil stabilization. Build. Environ. **42**(9), 3393–3398 (2007)
27. Solanki, P., Khoury, N., Zaman, M.: Engineering behavior and microstructure of soil stabilized with cement kiln dust. In Soil improvement, Geo-Denver **2007**, 1–10 (2007)
28. Yadu, L., Tripathi, R.K.: Effects of granulated blast furnace slag in the engineering behaviour of stabilized soft soil. Procedia Engineering. **51**, 125–131 (2013)
29. Singh, P., Dash, H.K., Samantaray, S.: Effect of silica fume on engineering properties of expansive soil. Materials Today: Proceedings. **33**(8), 5035–5040 (2020)
30. Sathiparan, N.: Utilization prospects of eggshell powder in sustainable construction material–A review. Constr. Build. Mater. **293**, 123465 (2021)
31. Raja, K., Venkatachalam, S., Vishnuvardhan, K., Krishnan, R.S.R., Selvan, V.T., Vetriselvan, N.A.: Review on soil stabilization using rice husk ash and lime sludge. Mater. Today: Proc. **65**(2), 1205–1212 (2022)
32. Ayub, F., Khan, S.A.: An overview of geopolymer composites for stabilization of soft soils. Constr. Build. Mater. **404**, 133195 (2023)
33. Taleb Bahmed, I., Harichane, K., Ghrici, M., Boukhatem, B., Rebouh, R., Gadouri, H.: Prediction of geotechnical properties of clayey soils stabilised with lime using artificial neural networks (ANNs). Int. J. Geotech. Eng. **13**(2), 191–203 (2019)
34. Mojtahedi, F.S.F., Ahmadihosseini, A., Sadegh, H.: An artificial intelligence based data-driven method for forecasting unconfined compressive strength of cement stabilized soil by deep mixing technique. Geotech. Geol. Eng. **41**, 491–514 (2023)
35. Gajurel, A., Mukherjee, P.S., Chittoori, B.: Estimating optimal additive content for soil stabilization using machine learning methods. In Eighth International Conference on Case Histories in Geotechnical Engineering, pp. 662–672. Reston, VA: American Society of Civil Engineers (2019)
36. Iqbal, M., Onyelowe, K.C., Jalal, F.E.: Smart computing models of California bearing ratio, unconfined compressive strength, and resistance value of activated ash-modified soft clay soil with adaptive neuro-fuzzy inference system and ensemble random forest regression techniques. Multiscale and Multidiscip. Model. Exp. and Des. 4, 207–225 (2021)
37. Mozumder, R.A., Laskar, A.I.: Prediction of unconfined compressive strength of geopolymer stabilized clayey soil using artificial neural network. Comput. Geotech. **69**, 291–300 (2015)
38. Khatti, J., Grover, K.S.: A scientometrics review of soil properties prediction using soft computing approaches. Arch Computat Methods Eng. (2023)
39. Khatti, J., Grover, K.S.: Prediction of UCS of fine-grained soil based on machine learning part 2: comparison between hybrid relevance vector machine and Gaussian process regression. Multiscale and Multidiscip. Model. Exp. and Des. **7**, 123–163 (2024)

40. Whang, S.E., Roh, Y., Song, H., Lee, J.G.: Data collection and quality challenges in deep learning: A data-centric ai perspective. VLDB J. **32**, 791–813 (2023)
41. Das, S.K., Samui, P., Sabat, A.K.: Application of artificial intelligence to maximum dry density and unconfined compressive strength of cement stabilized soil. Geotech. Geol. Eng. **29**, 329–342 (2011)
42. Narendra, B.S., Sivapullaiah, P.V., Suresh, S., Omkar, S.N.: Prediction of unconfined compressive strength of soft grounds using computational intelligence techniques: A comparative study. Comput. Geotech. **33**(3), 196–208 (2006)
43. Suman, S., Mahamaya, M., Das, S.K.: Prediction of maximum dry density and unconfined compressive strength of cement stabilised soil using artificial intelligence techniques. Int. J. of Geosynth. and Ground Eng. **2**, 11 (2016)
44. Chhabra, R.S., Mahadeva, R., Ransinchung, G.D.: Unconfined compressive strength prediction of recycled cement-treated base mixes using soft computing techniques. Road Mater. Pavement Des. **25**(2), 423–437 (2024)
45. Ghanizadeh, A.R., Heidarabadizadeh, N., Bayat, M., Khalifeh, V.: Modeling of unconfined compressive strength and Young's modulus of lime and cement stabilized clayey subgrade soil using Evolutionary Polynomial Regression (EPR). Int. J. Min. Geo-Eng. **56**(3), 257–269 (2022)
46. Pham, V.N., Do, H.D., Oh, E., Ong, D.E.: Prediction of unconfined compressive strength of cement-stabilized sandy soil in Vietnam using artificial neural networks (ANNs) model. Int. J. Geotech. Eng. **15**(9), 1177–1187 (2021)
47. Tinoco, J., Alberto, A., da Venda, P., Gomes Correia, A., Lemos, L.: A novel approach based on soft computing techniques for unconfined compression strength prediction of soil cement mixtures. Neural Comput. & Applic. **32**, 8985–8991 (2020)
48. Ochepo, J.: Stabilization of laterite soil using reclaimed asphalt pavement and sugarcane bagasse ash for pavement construction. J. Eng. Res. **2**(4), 1–13 (2014)
49. Gupta, G., Sood, H., Gupta, P.K.: Stabilizing Different Subgrade Soils with Pond Ash to Lower Greenhouse Gas Emissions for Bituminous Pavements in India, pp. 239–254. Urban Air Quality Monitoring, Modelling and Human Exposure Assessment (2021)
50. Heshmati, R.A.A., Alavi, A.H., Keramati, M., Gandomi, A.H.: A radial basis function neural network approach for compressive strength prediction of stabilized soil. In: Road Pavement Material Characterization and Rehabilitation: Selected Papers from the 2009 GeoHunan International Conference, pp.147–153 (2009)
51. Güllü, H.: Function finding via genetic expression programming for strength and elastic properties of clay treated with bottom ash. Eng. Appl. Artif. Intell. **35**, 143–157 (2014)
52. Ghorbani, A., Hasanzadehshooiili, H.: Prediction of UCS and CBR of microsilica-lime stabilized sulfate silty sand using ANN and EPR models; application to the deep soil mixing. Soils Found. **58**(1), 34–49 (2018)
53. Khan, K., Ashfaq, M., Iqbal, M., Khan, M.A., Amin, M.N., Shalabi, F.I., Faraz, M.I., Jalal, F.E.: Multi expression programming model for strength prediction of fly-ash-treated alkali-contaminated soils. Materials. **15**(11), 4025 (2022)
54. Taleb Bahmed, I., Khatti, J., Grover, K.S.: Hybrid soft computing models for predicting unconfined compressive strength of lime stabilized soil using strength property of virgin cohesive soil. Bull. Eng. Geol. Environ. **83**, 46 (2024)
55. Krishna, S.V., Santosh, B.S., Prasanth, B.S.: Prediction of UCS and CBR of a stabilized Black-cotton soil using artificial intelligence approach: ANN. Mater. Today: Proc. (2023)
56. Parthiban, D., Vijayan, D.S., Koda, E., Vaverkova, M.D., Piechowicz, K., Osinski, P., Van Duc, B.: Role of industrial based precursors in the stabilization of weak soils with geopolymer–A review. Case Stud. Constr. Mater. **16**, e00886 (2022)
57. Nagaraju, T.V., Mantena, S., Gobinath, R., Bonthu, S., Subhan Alisha, S.: Geopolymer-stabilized soils: influencing factors, strength development mechanism and sustainability. J. Taibah Univ. Sci. **17**(1), 2248651 (2023)
58. Bhavita Chowdary, V., Ramanamurty, V., Pillai, R.J.: Experimental evaluation of strength and durability characteristics of geopolymer stabilised soft soil for deep mixing applications. Innov. Infrastruct. Solut. **6**, 1–10 (2021)

59. Bozyigit, I., Zingil, H.O., Altun, S.: Performance of eco-friendly polymers for soil stabilization and their resistance to freeze–thaw action. Constr. Build. Mater. **379**, 131133 (2023)
60. Wang, S., Xue, Q., Zhu, Y., Li, G., Wu, Z., Zhao, K.: Experimental study on material ratio and strength performance of geopolymer-improved soil. Constr. Build. Mater. **267**, 120469 (2021)
61. Zeini, H.A., Al-Jeznawi, D., Imran, H., Bernardo, L.F.A., Al-Khafaji, Z., Ostrowski, K.A.: Random forest algorithm for the strength prediction of geopolymer stabilized clayey soil. Sustainability. **15**(2), 1408 (2023)
62. Rezazadeh Eidgahee, D., Rafiean, A.H., Haddad, A.: A novel formulation for the compressive strength of IBP-based geopolymer stabilized clayey soils using ANN and GMDH-NN approaches. Iran. J. Sci. Technol., Trans. Civ. Eng. **44**(1), 219–229 (2020)
63. Soleimani, S., Rajaei, S., Jiao, P., Sabz, A., Soheilinia, S.: New prediction models for unconfined compressive strength of geopolymer stabilized soil using multi-gen genetic programming. Measurement **113**, 99–107 (2018)
64. Javdanian, H., Lee, S.: Evaluating unconfined compressive strength of cohesive soils stabilized with geopolymer: a computational intelligence approach. Eng. Comput. **35**, 191–199 (2019)
65. Mozumder, R.A., Laskar, A.I., Hussain, M.: Empirical approach for strength prediction of geopolymer stabilized clayey soil using support vector machines. Constr. Build. Mater. **132**, 412–424 (2017)

A Review of Deformations Prediction for Oil and Gas Pipelines Using Machine and Deep Learning

Bruno S. Macêdo, Tales H. A. Boratto, Camila M. Saporetti, and Leonardo Goliatt

Abstract Pipelines are fundamental for conveying oil and gas, but deteriorations can have many effects, such as pollution and property deformities. Therefore, preserving pipeline integrity is important for a secure and sustainable energy provider. The fast development of Machine Learning (ML) methods gives a beneficial possibility to build predictive models that can efficiently resolve these complex problems. This review paper principally emphasizes applying deep learning (DL) and ML methods for estimating pipeline deformations in oil and gas production. The paper analyzes studies in this area, proposing a consistent discussion and determining the reasons and difficulties related to employing ML and DL for estimating deformations in pipelines. This review describes characteristics of ML algorithms often applied, presenting arguments. Based on a review, it is found that ML and DL procedures can precisely predict oil and gas pipeline deformations compared to typical techniques. Assessing ML predictive methods employed on reservoir or synthetic data for pipeline deformations can assist in developing monitoring tools for pipelines, reducing cost and time. It is expected that the development of this review will help

Please note that the AISC Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

B. S. Macêdo
Systems and Automation Engineering Graduate Program, Federal University of Lavras, Lavras, MG 37200-000, Brazil

T. H. A. Boratto
Computational Modeling Graduate Program, Federal University of Juiz de Fora, Juiz de Fora, MG 36036-900, Brazil

C. M. Saporetti (✉)
Department of Computational Modeling, Polytechnic Institute, Rio de Janeiro State University, Nova Friburgo, RJ 28625-570, Brazil
e-mail: camila.saporetti@iprj.uerj.br

L. Goliatt
Department of Computational and Applied Mechanics, Federal University of Juiz de Fora, Juiz de Fora, MG 36036-900, Brazil

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_16

comprehend the existing research gaps and give alternatives for other people who want to predict oil and gas pipeline deformations.

Keywords Deformation predict · Machine learning · Deep learning · Oil and gas pipelines

1 Introduction

Oil and gas are fundamental resources for the population, being responsible for providing about 85% of total traded energy. The request for oil is set to range at a level of 110.2 million barrels per day (bpd) in 2028, which represents an increase of approximately 11% regarding 2022. For 2045 the prevision is 16% [1]. Pipelines are essential to transport oil and gas, being an important part of the energy infrastructure, as they are responsible for providing these fuels leading from upstream zones to downstream end users. Therefore, the trust and security of these pipelines are necessary [2].

Oil and gas pipelines are perceived as one of the best means of transport considering security. Despite that, the statistics point to a rising tendency in the number of incidents [3]. In addition, a pipeline deformation can result in grave lesions in the people, financial losses, and nature destruction. The Pipeline and Hazardous Materials Safety Administration (PHMSA) of the United States manages information about the rate of pipeline failures that are due to deformations. Considering the incidents occurring in 2022, the majority were generated by material failure (and similar), fret, and excavation deteriorate [4]. Consequently, pipeline structural physical conditions observing and building of models are essential for preventing these unwelcome adverse results occasioned by incidents of pipeline deformations.

Survey strategies have been employed to find pipeline irregularities and defects without paralyzing down producing. With the aim of decreasing the time and price spent on conventional techniques, studies have been carried out to determine deformation and failures causes. In this context, methodologies that use Machine Learning (ML) and Deep Learning (DL) techniques to estimate deformations have been proposed in the literature.

ML and DL methods can be employed to estimate pipeline deformations caused by evaluating many characteristics of the pipeline itself, natural circumstances, and geographical area [5]. By exploring these elements, these methods can determine relations and similarities that people may not be able to identify, permitting an increase in precision estimations of pipeline deteriorations. Also, ML and DL can be applied to make pipeline preservation more efficient through the prediction of locations that are more susceptible to damage, enabling preventive maintenance and reducing the chance of damage. Furthermore, these approaches can be employed to observe pipelines instantly, determining feasible deteriorations once it happens and notifying workers to act accordingly.

Considering the diversity of the ML and DL methods, it is necessary to conduct a literature review to differentiate these approaches, showing their strengths and limitations in estimating deformations. A review could permit a link between researchers and people acting in the practice of damaged oil and gas pipeline prediction by informing them about the literature contributions and progress on this theme. This paper aims to introduce a brief review of research existing on deformations of oil and gas pipeline prediction methods using ML and DL techniques targeted.

The remainder of this paper is divided as follows: Sect. 2 presents an explanation of machine learning and deep learning along with the algorithms covered in this work. Section 3 the methodology used. Section 4 presents and discusses a review of deformation and corrosion prediction for oil and gas pipelines. Section 5 covers a discussion, limitations, challenges, and future perspectives and the conclusions are in Sect. 6.

2 Machine Learning and Deep Learning Concepts

2.1 Machine Learning Algorithms

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that builds computational algorithms, using mathematical and statistical techniques, learning based on a pre-defined database, at the end generating a model for classification, detection or prediction [6]. Developing an ML algorithm consists of three steps: pre-processing, training, and model evaluation. In the first step, the database is organized, the problem you want to solve is defined and the training and testing data is divided. ML model training can occur in a supervised and unsupervised manner [7–10].

In supervised learning, model training is carried out using a set of data with the true label already assigned, while unsupervised learning occurs based on information without the correct label information [9]. In the evaluation stage, the predicted output is acquired with the results generated by the model. Therefore, ML models learn through repeated observations and establish a pattern to create a model capable of generalizing the information, so that new data is labeled accurately [10]. It is important to highlight that in the development stage of an ML algorithm, a consolidated database must be used to avoid generating misleading results [11].

Supervised machine learning consists of using a database in which the data is labeled through the knowledge of a professional in the field [12]. In this learning process, a rule is inferred to assign a class to a never-before-seen object from training data previously classified. Supervised learning predicts an output for each input object based on the pattern learned in each of the classes in training carried out with data labeled [13].

Unsupervised learning uses a database in which the data is unlabeled. In this way, it seeks to find a pattern and determine how the data is organized [13]. It is often used in cases where there is a large amount of data, and labeling this data manually is not a

trivial task. Unsupervised learning techniques include clustering and dimensionality reduction, among others.

2.1.1 Artificial Neural Network

Artificial Neural Networks (ANN) are machine learning methods that are based on the human brain. Through ANN, it is possible to carry out the learning, organization, and generalization capacity of data [14]. Artificial neurons are the smallest units of ANN.

A set of different neurons connected in layers form a network. The communication of neurons in this network occurs through synaptic weights. Figure 1 shows an example of an artificial neuron model.

In the ANN learning process, an adjustment of the synaptic weights is carried out (wi), seeking to minimize the error in the network prediction. Because of the ability to faithfully model non-linear actions, ANNs are important tools for controlling systems [15]. An ANN has different components, including:

- **Set of weights**: connections between ANN neurons. Each of the neurons has a weight;
- **Integrator**: adds the ANN input signals, weighted by the weights;
- **Activation function**: limits the value of a neuron obtained at the output;
- **Bias**: value applied to each neuron with the idea of increasing or decreasing the value of the net input of the function activation.

Learning a neural network is a process in which parameters are changed in a learning process in the network environment. ANNs are composed of a certain number of inputs and processing units, which are connected through synaptic weights. Inputs are propagated through the topology of the ANN, in which they are transformed by the synaptic weights and the activation function (AF) of neurons. Given inputs from n neurons (y_i), the neuron k performs the output calculation through Eq. 1.

$$Y_k = AF\left(\sum_{i=1}^{n}(y_i \cdot w_{ki}) + b_k\right) \tag{1}$$

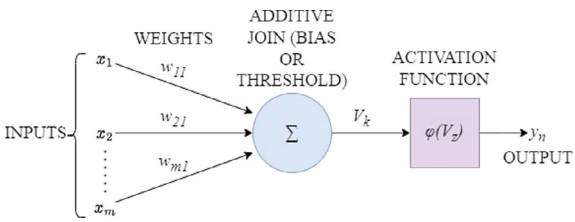


Fig. 1 Model of an artificial neuron

where y_i is the output calculated by neuron i, w_{ki} is the synaptic weight between neuron i and the k and b_k the neuron represents the weight between a value constant and non-zero to neuron k, known as bias. If the neuron is linked to the inputs, the term y_i is replaced by the corresponding entry.

In an ANN it is necessary to determine the weights synaptic and bias to use. To estimate such parameters, training is carried out, constituting an iterative process where the initial parameters are used until the process converges. Considering interaction j, the weight w_{ki} is used through Eq. 2.

$$w_{(j)ki} = w_{(j-i)ki} + \delta w_{(j)i} \tag{2}$$

where $\delta w_{(j)i}$ represents the correction vector to the w_{ki} parameter in iteration j.

According to Haykin [12], the use of neural networks offers nonlinearity, adaptability, natural language processing, voice recognition, and image classification, among other factors. There are several ANN models used in the literature, such as the Multilayer Perceptron Neural Networks (MLP) and Extreme Machine Learning (ELM).

2.1.2 Decision Tree

A Decision Tree (DT) is a supervised method that has a tree structure, where the data set is divided into smaller sets, already a decision tree that is associated, its development is incremental. In the end, we obtain a tree with decision nodes and leaf nodes. A DT takes a set of properties as input and returns a result that is the expected class of the input values. The decision is made by running a series of tests. Each internal node in the tree corresponds to a feature value test, and branches from that node identify possible test values. Each leaf node performs a specification of the value returned if the leaf node is accessed [16].

A DT uses the strategy called divide and conquer, in which an abstract problem is decomposed into smaller and simpler subproblems. Likewise, this strategy is also applied to each subproblem [17]. When using DT, there are some advantages: they do not assume any particular distribution for the data; characteristics can be categorical (qualitative) or numerical (quantitative); any function can have the construction of methods, this is only possible if the number of examples for learning is sufficient. Some of the DT algorithms are Inductive Decision Trees (ID3), Classification and Regression Trees (CART), and Chi Square Automatic Interaction Detection (CHAID), among others [18].

Figure 2 illustrates an example of a decision tree in the context of material deformation. Each rectangle represents a deformation class (low, medium, or high). To classify the deformation condition of a material, simply start at the root, following a condition until you reach a leaf.

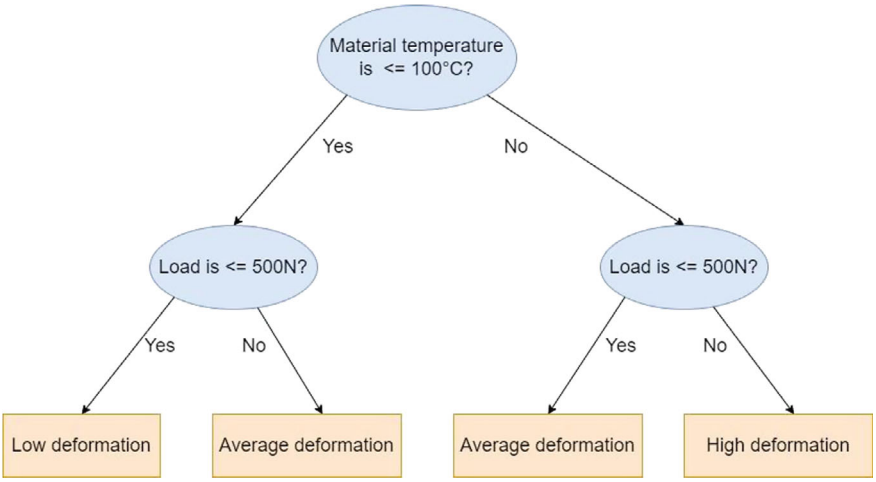


Fig. 2 An example of a simple decision tree in the context of the deformation condition of a material

2.1.3 Support Vector Machine

The Support Vector Machine (SVM) is a supervised machine learning model, with the objective of creating a division between two or more classes, allowing classes to be predicted based on feature vectors [19]. The decision barrier is called a hyperplane, for better prediction, the further it is from the closest data points in each of the classes, the better the prediction will be. These points at which closest to [20] are called support vectors. Let be a labeled training data set expressed by Eq. 3.

$$(x_1, y_1), \ldots, (x_n, y_n), x_i \in R^d,\ y_i \in (-1, +1) \tag{3}$$

where x_i indicates a feature vector, y_i the class, which can be negative or positive of a training item i. The optimization of the hyperplane is represented by Eq. 4.

$$w^T x + b = 0 \tag{4}$$

where w is the weight vector, x is the input feature vector, and b is the bias. w and b must satisfy the inequalities present by Eq. 5 for the entire training set.

$$\begin{aligned} w^T x + b \geq 1, if y_i = 1 \\ w^T x + b \leq 1, if y_i = -1 \end{aligned} \tag{5}$$

SVM training aims to find w and b that allow the separation of data in the hyperplane so that the margin is maximized.

2.1.4 Ensemble

Ensemble models are based on the combination of results obtained by a set of classifiers. The main objective of these methods is to have higher classification performance and lower variance compared to individual classification models [21]. Models generated by the same technique may differ from each other. Aiming to expand forecasting capacity, a methodology that has been increasingly explored is the use of different machine learning algorithms to carry out joint forecasting, a technique called Ensemble. There are different Ensemble techniques that can be used, such as:

1. **Bagging**: which generally consists of training homogeneous models in parallel and combining their results through a deterministic technique to generate the final prediction [22];
2. **Boosting**: trains homogeneous models sequentially. The weighted sum of the trained models results in a final model that encompasses the characteristics treated in each model individually [23];
3. **Stacking**: is a technique that uses a set of normally heterogeneous models, trained in parallel and a metamodel that is trained with the set of predictions made by the first layer models and performs the final prediction of the problem [24].

Figure 3 presents the structure of an Ensemble with k components and a combiner method, in which each component produces an output for the same input x.

2.2 *Deep Learning Algorithms*

Deep learning consists of several Machine Learning techniques that use deep neural networks between the input and output layers with many intermediate layers [25]. Machine learning algorithms with varying degrees of abstraction are used, working at different levels. The learned algorithms correspond to different levels of concepts, where high-level concepts are defined from lower levels, and the same lower level concepts help define many higher level concepts focusing on learning of multiple levels of representation and abstraction that help make sense of data. Deep Learning algorithms aim to discover a method based on a set of data that will serve as a guide for learning the model from these examples [26].

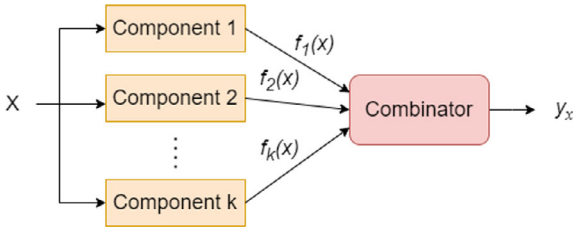


Fig. 3 Structure of an ensemble with k components and a combiner method

Some of the concepts that stand out about Deep Learning in the literature are models that consist of multiple layers or stages of nonlinear information processing, and methods for supervised or unsupervised learning of feature representation in consecutively higher and more abstract layers. Deep learning stands out for improving information processing capacity, increasing the size of data used in training, and current advances in research in machine learning and signal processing, and these advances have enabled such methods to effectively analyze complex and non-linear functions, learn representations of distributed and hierarchical resources [27, 28].

Deep networks for unsupervised, supervised, and hybrid learning are the three main workspace groups of deep learning. Unsupervised learning is based on capturing the high-order correlation of visible data to analyze patterns when there is no information about the labels of the classes present in the base. Supervised learning of deep networks is intended to directly provide information for the purpose of classifying patterns, often characterizing the posterior distributions of classes conditioned on visible data. The target label data is always available directly or indirectly for this supervised learning. They are also called discriminative deep networks. Hybrid deep networks stand out for the discrimination that is assisted, often meaningfully, with the results of generative or unsupervised deep networks. This can be achieved through better optimization and regularization of supervised deep networks. The proposal of this network can also be achieved when discriminative criteria for learning supervised networks are used to estimate the parameters in any of the deep generative or unsupervised [26, 29, 30].

2.2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of deep neural network well-used in computer vision tasks. CNNs have as their idea the hierarchical organization of the human visual cortex. As in ANN, CNNs also have artificial neurons to act in the learning process, feature extraction, and input segmentation. CNNs have convolutional layers, pooling layers, and dense layers [31, 32].

In the convolutional layer, filters are defined for the convolution step to manipulate the input images or maps from the previous layer into feature maps in the output layer. In neurons that are in the convolution layers, convolution filters are applied to the input image or the outputs of the previous layer and as a result, the features are learned, this information is stored in the layer's output in a feature map. The convolutional layers and convolution filters are applied in a small region of the input set to compose specific information about the component, and color in an RGB image, being treated. In this process, an activation function with non-linear behavior is applied. Some activation functions used are the hyperbolic tangent, rectified linear function (ReLu), Softmax, and the sigmoid. The feature maps obtained after using the activation function are the features obtained and are input information for the following layers [31, 32].

The grouping layer is used after the convolution layer and applies downsampling by reducing the size of the input data of the next layer. One idea of a clustering

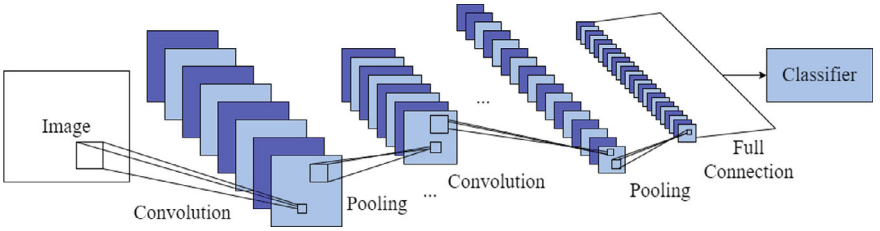


Fig. 4 Example of layers of a CNN network

function is the representation of a given clustering window by its largest amplitude (max poling). Considering a 2 × 2 dimension grouping, the function returns the largest value among the four analyzed. In the dense layer, each neuron is connected to all the neurons in the previous layer. An evaluation of the features that are extracted is carried out, generating a probability vector of N dimensions, in which N represents the number of outputs in the dense layer. After this layer, an activation function, such as sigmoid or SoftMax, is used to generate the probability for classifying the data set [31, 32].

In CNN training, the idea is to minimize the loss function, indicating the network error, in which stochastic gradient descent is used as an approach. In the first stage, the information is processed in the network, calculating the output. In the second stage, the error between the real values and those obtained from the network is calculated. Then, backpropagation and minimization of this error are performed, adjusting the precision matrix and the network [33, 34].

Figure 4 illustrates an example of layers of a CNN network that at the end performs a classification.

2.2.2 Autoencoder

Autoencoder (Autocoding Networks or Autoassociative Networks) is a type of neural network that seeks to find a smaller way of representing input data [26], due to the fact that they generally have as output the input they receive. In training, there are some restrictions for learning significant information from the data set, and not using only an identity function in training. The idea of representing them is through an Encoder network and a Decoder network, in which they are trained together, but for a given application they are not used together. The encoder is responsible for receiving an input set x so that information r is generated. The decoder is responsible for receiving information from the data set r, to generate a reconstruction $\hat{x}$ of the data set. This idea is represented by Eqs. 6 and 7.

$$r = Enc(x, \vartheta) \tag{6}$$

$$\hat{x} = Dec(r, \varphi) \tag{7}$$

One way for the autoencoder to generate relevant information is through the use of Denoising Autoencoders: in the training stage, the input is changed by outliers and the expected output is the original input without change. To solve this problem, the autohinder needs to learn about the probabilistic distribution from which the inputs are sampled. Therefore, the objective of the training is to learn distribution [26].

2.2.3 Recurrent Neural Network

Recurrent Neural Networks (RNN) are a type of ANN in which inputs are fed back, causing them to have temporal dynamic behavior. Neurons are connected with neurons from previous layers, or the same layer. Thus, information does not happen in a single direction, nor does the network output depend only on the current input, but on previous inputs. The consequence of this is the existence of short-term memory in the RNN [12]. The RNN network has the idea of short-term memory, in which it remembers information from the past.

Considering a network in which a vector $x(n) \in R^p$ is the input vector and the vector $v(n) \in R^q$ indicates the output of the layer hidden in time n, so the dynamic mode of the coupled-pair RNN can be explained by Eqs. 8 and 9.

$$v(n+1) = \varphi(v(n), x(n) \tag{8}$$

$$y(n) = \varphi(v(n)) \tag{9}$$

in which $\varphi(.)$ indicates a non-linear function, characterizing the hidden and output layer [35].

One way to classify RNNs is by the involvement of neurons from just one layer or multiple layers, that is, concerning the feedback connection. It can be defined into three groups: local recurring connection, global recurring connection, and non-local recurring connection. In a local recurring connection, the connection involves only one neuron. The term local is because the same neuron is fed back by its output. Global recurring connection a neuron from a given layer is connected to a neuron from the previous layer, that is, the value obtained at the output of a neuron is fed back as input to a neuron from the previous layer. Finally, non-local recurring connection involves many different neurons, but these neurons are connected to neurons in the same layer. The output of a given neuron is fed back as input to a different neuron in the same layer.

2.2.4 Long Short Term Memory

Long Short-Term Memory (LSTM) networks are a type of RNN, with the idea of solving the long-term dependency problem of RNNs. This problem, known as the Vanishing Gradient, concerns relevant information that is lost in the network when

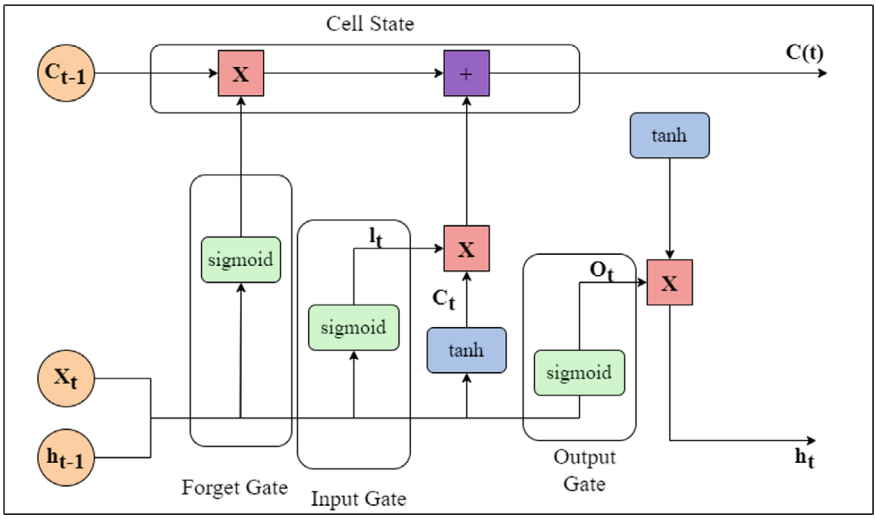


Fig. 5 Architecture of an LSTM

it works with a large sequence of data. In this sense, the network seeks to remember information from large periods of time [36, 37].

In an LSTM architectural pattern, a memory cell approach is used, allowing the network to remember information after several iterations. It also has valves with three multiplicative units: input valve, output valve, and forget valve. The valves have the the objective of preserving the constant error flow in units called gates, making it possible to adjust the weights such as truncation of the sequence of unimportant information, meaning forgetting [37, 38].

Each element present internally in an LSTM network has a purpose. Cell State has the most fundamental function of the LSTM core. It allows information that exists through linear interactions to travel through the network over time. The forget gate decides whether the information that is obtained from the previous state will be kept or discarded. Input gate makes the decision on the information that is obtained from the input that is inserted and combined with the previous ones. The output gate is responsible for determining the parts that are sent at the output [37]. Figure 5 illustrates the architecture of an LSTM.

3 Methodology

In order to carry out this review, one used the Scopus database to encounter pertinent papers. Distinct search strings were employed involved deformation or corrosion prediction more a machine learning or deep learning method and oil or gas pipeline, and the interval was set from 2013 to 2023.

Papers that meet the exclusion criteria listed below will not be considered:

– do not use machine learning or deep learning methods;
– does not address deformation or corrosion in oil and gas pipelines;
– duplicates, that is, those that appeared in more than one search.

The search strings used and the results obtained are exhibited in Tables 1 and 2 for deformation and corrosion, respectively. It can be observed that the searches return more papers through machine learning than through deep learning methods. Among the methods, artificial neural networks returned the most. The number of papers returned using corrosion is greater than deformation. Despite returning a relatively high number of papers, when analyzed, few are related to the problem in question.

Table 1 Search strings and no. research papers of deformation

String Search	Machine learning	No. Selected Papers
	No. Research Papers	
Deformation oil pipelines AND Artificial Neural Network	273	7
Deformation gas pipelines AND Artificial Neural Network	371	5
Deformation oil pipelines AND Decision Tree	56	5
Deformation gas pipelines AND Decision Tree	72	1
Deformation oil pipelines AND Support Vector Machine	124	7
Deformation gas pipelines AND Support Vector Machine	172	0
Deformation oil pipelines AND Ensemble	83	0
Deformation gas pipelines AND Ensemble	103	1
	Deep Learning	
String Search	No. Research Papers	No. Selected Papers
Deformation oil pipelines AND Convolution Neural Network	21	0
Deformation gas pipelines AND Convolution Neural Network	31	0
Deformation oil pipelines AND Autoencoder	16	0
Deformation gas pipelines AND Autoencoder	16	0
Deformation oil pipelines AND Recurrent Neural Network	30	0
Deformation gas pipelines AND Recurrent Neural Network	40	1
Deformation oil pipelines AND Long short-term memory	35	0
Deformation gas pipelines AND Long short-term memory	39	0

Table 2 Search strings and no. research papers of corrosion

	Machine learning	
String Search	No. Research Papers	No. Selected Papers
Corrosion oil pipelines AND Artificial Neural Network	840	12
Corrosion gas pipelines AND Artificial Neural Network	966	10
Corrosion oil pipelines AND Decision Tree	307	10
Corrosion gas pipelines AND Decision Tree	346	6
Corrosion oil pipelines AND Support Vector Machine	377	8
Corrosion gas pipelines AND Support Vector Machine	462	4
Corrosion oil pipelines AND Ensemble	239	6
Corrosion gas pipelines AND Ensemble	276	2
	Deep Learning	
String Search	No. Research Papers	No. Selected Papers
Corrosion oil pipelines AND Convolution Neural Network	241	11
Corrosion gas pipelines AND Convolution Neural Network	279	4
Corrosion oil pipelines AND Autoencoder	43	3
Corrosion gas pipelines AND Autoencoder	48	0
Corrosion oil pipelines AND Recurrent Neural Network	86	3
Corrosion gas pipelines AND Recurrent Neural Network	100	1
Corrosion oil pipelines AND Long short-term memory	83	6
Corrosion gas pipelines AND Long short-term memory	89	8

4 Review Deformation and Corrosion Prediction for Oil and Gas Pipelines

4.1 Machine Learning

Figures 6 and 7 show the results of the search on machine learning and deformation. It can be seen that ANN and SVM return a greater number of papers, analyzing the number of research papers found and the number of selected papers according to the strings of the search. Regarding the year, 2023 followed by 2022 was when the most papers were published, considering the problem in question. Figures 8 and 9

present the results of the search on machine learning and corrosion. A larger number of papers were returned for all search strings compared to deformation. Despite this, a few papers were selected when considering the exclusion criteria. Between the years 2021 and 2023, it was where more articles were published, increasingly in the period.

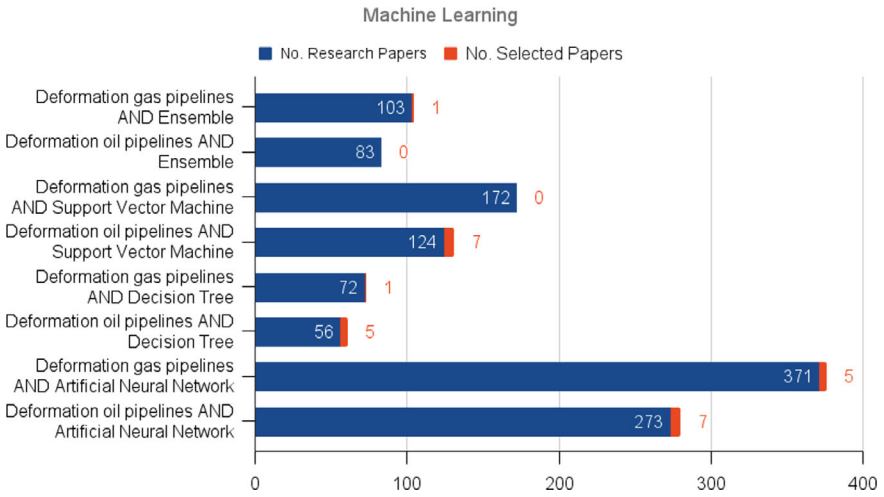


Fig. 6 No. Research Papers x No. Selected Papers—ML (Deformation)

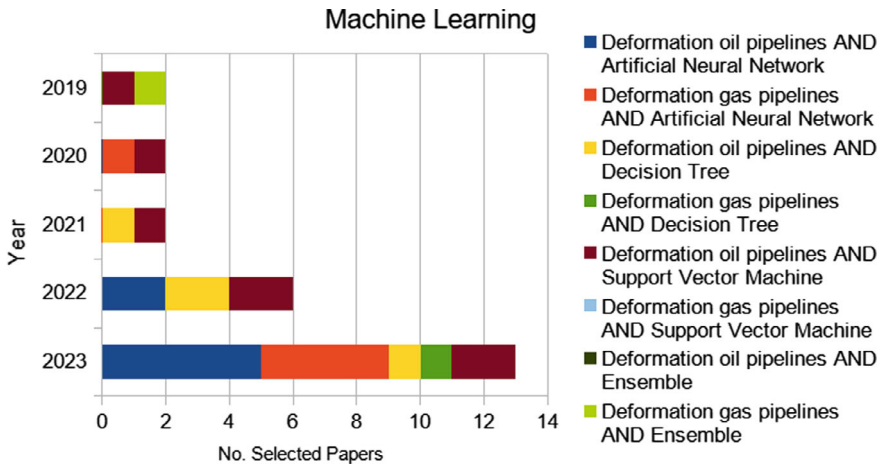


Fig. 7 No. Selected Papers x Year—ML (Deformation)

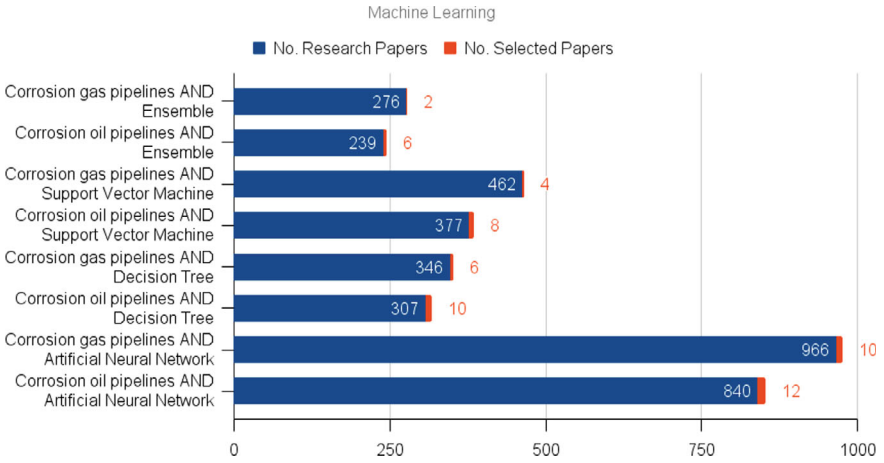


Fig. 8 No. Research Papers x No. Selected Papers—ML (Corrosion)

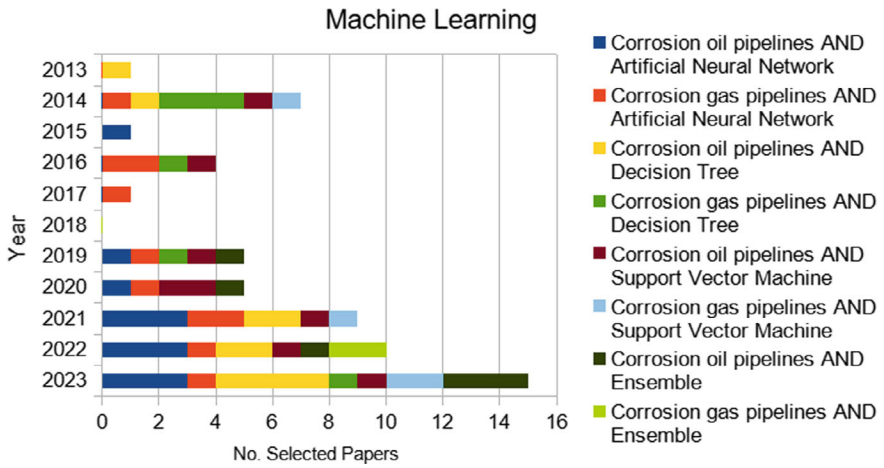


Fig. 9 No. Selected Papers x Year—ML (Corrosion)

4.2 Deep Learning

Figures 10 and 11 show the results of the search on deep learning and deformation. The number of papers found was lower than with machine learning, which shows that there are cases in which there is no need to use a more robust method that requires greater processing. Considering the deep learning methods used in the search strings, it is noted that there are more published works using recurring networks such as LSTM and RNN. However, after applying the exclusion criteria, it resulted in only

one published paper to predict deformation in oil and gas pipelines and this paper was published in 2022.

Considering corrosion and analyzing Figs. 12 and 13, it is clear that most of the studies found applied CNN, which was expected since corrosion in many cases is predicted through images. In this case, more papers were selected and were mostly published in 2022 and 2023.

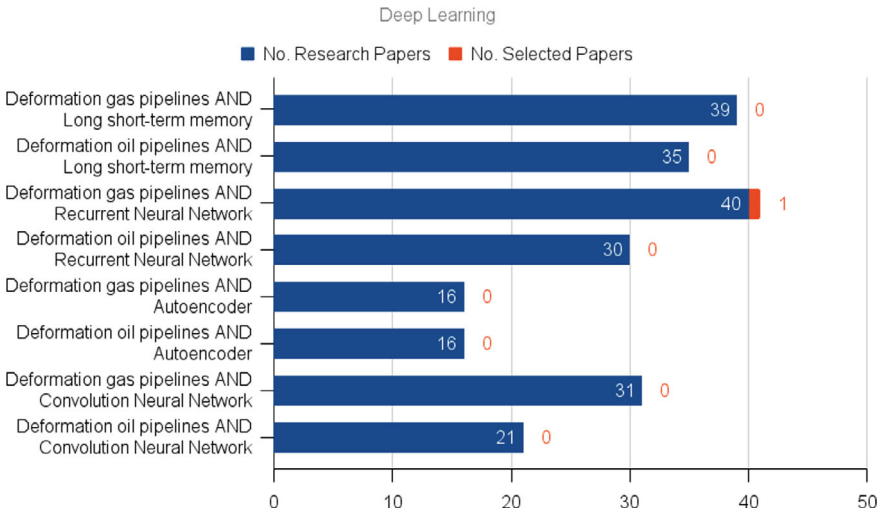


Fig. 10 No. Research Papers x No. Selected Papers—DL (Deformation)

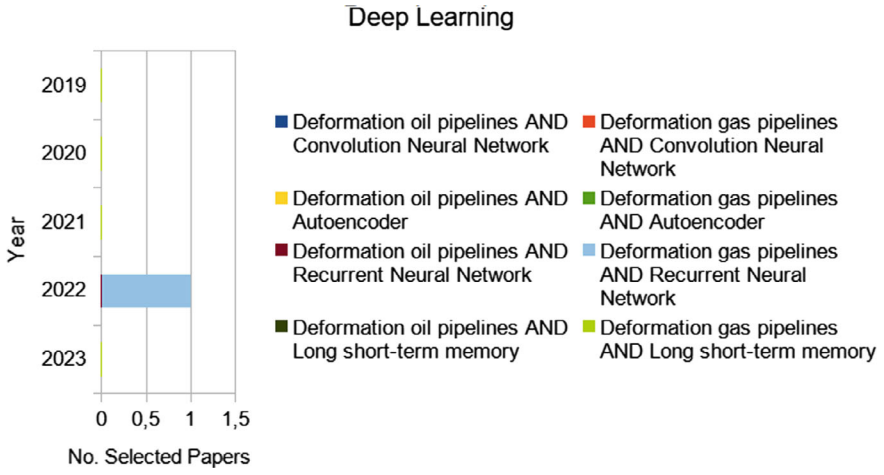


Fig. 11 No. Selected Papers x Year—DL (Deformation)

Fig. 12 No. Research Papers x No. Selected Papers—DL (Corrosion)

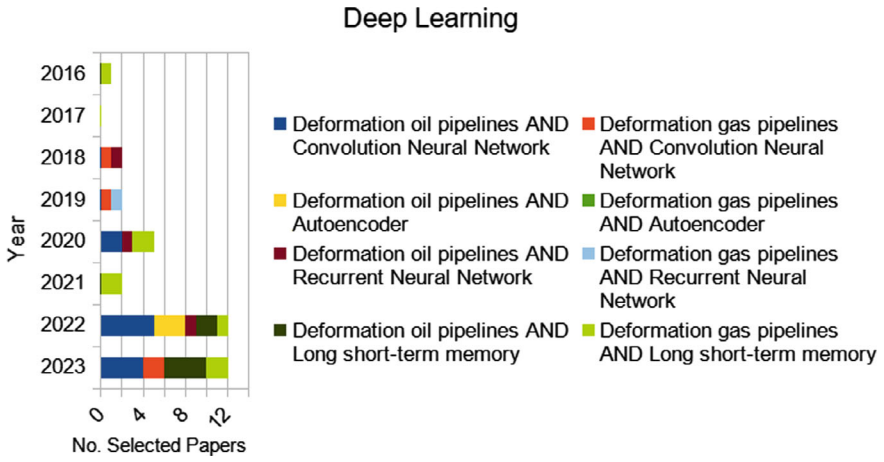


Fig. 13 No. Selected Papers x Year—DL (Corrosion)

5 Discussions: Strengths and Limitations, Challenges and Future Perspectives

Tables 3 and 4 present the ML and DL methods used in applications related to deformation prediction. It can be observed ANN and SVM were more employed in the references and the main applications are to identify deformations, detect pipeline leakage, and predict the failure pressure of corroded pipelines.

Table 3 References selected—ML (Deformation)

Methods	References	Main applications
SVM	[39–45]	Estimate the maximum pitting depth of buried pipeline; Identify the open joint, crack, and deformation defects
Fuzzy Bayesian Network	[46]	Characterize the corrosion risk
XGBoost	[42, 47, 48]	Predict corrosion fatigue life; Assess pipeline safety; Monitor the stress–strain state of pipelines
DT	[49]	Model the additional stress; Evaluate pipeline
ANN	[50–56]	Predict the failure pressure of corroded pipeline; Predict the fatigue life of the dented pipe; Estimate fatigue life capacity
AdaBoost	[42]	Model the additional stress
Random Forest	[42, 57]	Model the additional stress; Predict impact toughness
KNN	[57]	Predict impact toughness
BPNN	[58–60]	Calculate the reliability of corroded pipeline and the effects of axial compression force; Characterize the deformation
Mamdani fuzzy inference	[61]	Predict when the gas pipeline is hazarded
Ridgelet neural network	[62]	Predict the safety status of hydrogen pipeline

Table 4 References selected—DL (Deformation)

Methods	References	Main applications
Autoencoder	[44, 63]	Detect pipeline leakage
CNN	[64]	Detect of pipeline leaks under multiple working conditions
LSTM	[65]	Estimate the maximum pitting corrosion depth

Table 5 and 6 show the ML and DL methods used in applications related to corrosion prediction. It can be seen ANN, Bayesian network, RF, SVM, CNN, and LSTM were more used in the research. In addition, the main applications are to classify different natural gas pipeline failures and predict the corrosion rate.

Table 7 presents the machine learning and deep learning techniques, used in the search, advantages and disadvantages highlighted in the papers during this review. It can be seen that among the advantages one can highlight high accuracy, and the ability to deal with different data, and the disadvantages are the high computational cost and the possibility of overfitting.

The papers presented show a wide range of machine learning and deep learning methods, encompassing SVM, RF, ANN, CNN, LSTM. Approaches using these techniques are important for the development of tools that enable the prediction of

Table 5 References selected—ML (Corrosion)

Methods	References	Main applications
BPNN	[66–71]	Detect leakage; Predict corrosion failure; Identify the pipelines defects
SVM	[72–87]	Classifydifferentnatural pipeline failures; Predict of the corrosion rate
DT	[73, 86, 88, 89]	Classify pipeline defects
RF	[73, 79, 87, 88, 90–94]	Predict the internal corrosion rates; Classify pipeline defects
Non-linear least squares	[95]	Quantitative analysis of the risk and failure
Gradient Boosting Machine	[90, 92]	Predict the internal corrosion rates
Feed-Forward Artificial Neural Network	[90]	Provide information for maintenance and inspection planning
KNN	[74, 75, 87, 88, 96, 97]	Predict corrosion rate; Detect rate, false detection rate, and missed detection rate of defects
KernelExtremeLearning Machine (KELM)	[98]	Predict the depth and length of corrosion defects
CatBoost	[91]	Predict the internal corrosion rates
XGBoost	[87, 88, 92, 99]	Predict corrosion rate; Determinate the maximum corrosion depth
Bayesian network	[100–108]	Identify the critical pipelines in a high consequence area; Prediction of failure probability; Quantify the health state of the pipeline over time
ANN	[78, 79, 82, 86, 87, 89, 109–122]	Predict the condition of existing oil and gas pipelines; Estimate the failure cause of oil pipelines; Classify different natural gas pipeline failures
Fuzzy Petri Net	[123]	One can to verify that the risk evaluation in the long-distance oil and gas transportation pipelines
Multiple Linear Regression	[124–126]	Predict the condition of oil and gas pipelines based on several factors including corrosion
Fuzzy Information Fusion	[127]	Corrosion detection

Table 6 References selected—DL (Corrosion)

Methods	References	Main Applications
LSTM	[76, 128–133]	Classify and detect damage to pipes in different types of conditions and materials; Identify blockage and leakage states; Detect leakage
CNN	[93, 129, 134–145]	Determine the presence of a defect; Detect and classify corrosion; Identify the types of pipeline failure defects
Autoencoder	[97, 129, 131, 133, 146]	Detect leakages; Diagnose defects
Generative adversarial network	[147, 148]	Identify leakages; Pipeline integrity management
Generalization-Memorization model	[149]	Predict failure rates
RNN	[150, 151]	Predict the life of equipment with corrosion dimension classes exposed to critical condition monitoring
Deep Neural Network	[152–154]	Predict the failure pressure of defective pipelines; Detect leakage

deformations and corrosion in oil and gas pipelines quickly and efficiently. Such results can help professionals who carry out this monitoring, so that they can make more assertive decisions, contributing to damage reduction. These strategies perform better if feature selection and metaheuristics techniques or other techniques are previously applied to select the best parameters. Furthermore, the availability of data that represents the phenomenon well is something that influences the training process method and interferes with prediction.

Data might be hard to get by in some places, particularly in isolated or difficult-to-reach places. Furthermore, the accuracy of the forecasts may be jeopardized by problems with the data, such as noise. It is critical to solve these concerns of data availability and quality, focusing on more advanced and efficient techniques for data collecting and processing, in order to support the resilience of the model. Another significant obstacle to large-scale corrosion and deformation prediction in oil and gas pipelines is overfitting. This problem happens when models learn specifics about the training data by heart, rather than applying broad trends to areas outside of the training set. Creating machine learning and deep learning models that can apply newly learned knowledge to new data is essential to reducing this. This guarantees more accurate forecasts that work in a variety of situations.

More dependable and useful data have also been obtained as a result of recent efforts in uncertainty estimates and active learning, which has sped up the mapping process by decreasing the need for labeled data. Furthermore, the introduction of explainable AI methodologies has brought transparency to the decision-making

Table 7 Advantages and Disadvantages of ML and DL for deformation and corrosion prediction in oil or gas pipelines

Type	Method	Advantage	Disadvantage
Machine Learning	ANN DT SVM Ensemble	Ability to handle incomplete or noisy information Parallel Processing Can be useful with or without hard data, New options can be added to existing trees Learns concepts not belonging to the original data It is effective in problems whose database has high dimensionality Weak models become strong Helps capture non-linear relationship in data	Training time tends to be long. Difficulty in finding the optimal network architecture and any data requires minimal preparation Is prone to overfitting Unstable to Noise Tuning model hyperparameters requires multiple tests to find the best cost–benefit performance Is prone to overfitting Can be computationally expensive
	CNN Autoencoder	Automatic feature extraction. Highly accurate at image recognition and classification Can extract important features and reduce the noise or the useless features Helps in data compression	High computational requirements. Large memory footprint Enhances noise reduction Denoising process can result in loss of some information that is needed from the original input
Deep Learning	RNN LSTM	Ability to handle variable length sequences Memory of past inputs Handling long sequences Gradient flow control during backpropagation	Due to its recurrent nature, the computation is slow Training of RNN models can be difficult Training can be slower and may require more resources Have several hyperparameters to tune

process, offering insightful analyses of model predictions and bolstering acceptance and confidence in ML and DL-based deformation prediction tools.

Together, these noteworthy developments offer promising futures for resolving issues, bolstering security protocols, and the transfer of gas and oil. In conclusion, notwithstanding present difficulties, ML and DL will continue to contribute to predicting deformation in oil and gas pipelines. One can transcend present constraints and get more accurate, thorough, and dependable results by utilizing methodological and technical breakthroughs, opening the door to new discoveries and applications.

6 Conclusions

In this paper, a systematic literature review was carried out on deformation prediction models in oil and gas pipelines based on ML and DL. The investigation of these strategies has shown a panorama of developments and useful applications that are extremely promising. Remarkably successful automatic and accurate deformation prediction has been achieved by combining a variety of data (pictures, numbers, videos, etc.) with well-known machine learning and deep learning algorithms including SVM, ANN, RF, CNN, RNN, LSTM, Ensemble, and Autoenconder.

Finally, the techniques covered in this work offer priceless instruments for automated, precision-driven deformation prediction helping in a variety of fields, including risk assessment and pipeline monitoring. The success is carefully choosing the approach that best fits the goals of the application and the available data.

References

1. Telli, A.: Evolution of the energy security concept—new dimensions and challenges. Security Studies: Classic to Post-Modern Approaches, 153 (2023)
2. Cobanoglu, M.M., Kermanshachi, S., Damnjanovic, I.: Statistical modeling of corrosion failures in natural gas transmission pipelines. In: Pipelines 2016, pp. 195–204 (2016)
3. Mahmood, Y., Afrin, T., Huang, Y., Yodo, N.: Sustainable development for oil and gas infrastructure from risk, reliability, and resilience perspectives. Sustainability **15**(6), 4953 (2023)
4. Xiao, R., Zayed, T., Meguid, M.A., Sushama, L.: Understanding the factors and consequences of pipeline incidents: an analysis of gas transmission pipelines in the us. Eng. Fail. Anal. **152**, 107498 (2023)
5. Aljameel, S.S., Alomari, D.M., Alismail, S., Khawaher, F., Alkhudhair, A.A., Aljubran, F., Alzannan, R.M.: An anomaly detection model for oil and gas pipelines using machine learning. Computation **10**(8), 138 (2022)
6. Mitchell, T.M.: The Discipline of Machine Learning vol. 9. Carnegie Mellon University, School of Computer Science, Machine Learning …, ??? (2006)
7. Barreto, G.A., Souza, L.G.M.: Adaptive filtering with the self-organizing map: a performance comparison. Neural Netw. **19**(6–7), 785–798 (2006)
8. Kohonen, T., Honkela, T.: Kohonen network. Scholarpedia **2**(1), 1568 (2007)
9. Sathya, R., Abraham, A., et al.: Comparison of supervised and unsupervised learning algorithms for pattern classification. Int. J. Adv. Res. Artif. Intell. **2**(2), 34–38 (2013)
10. Rajkomar, A., Dean, J., Kohane, I.: Machine learning in medicine. N. Engl. J. Med. **380**(14), 1347–1358 (2019)
11. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mobile Netw. Appl. **19**, 171–209 (2014)
12. Haykin, S.: Redes Neurais: Princípios e Prática. Bookman Editora, ??? (2001)
13. Mitchell, T.M.: Machine Learning. McGraw-hill (1997)
14. Krose, B., Krose, B., Smagt, P., Smagt, P.: An introduction to neural networks (1993). http://citeseerx.ist.psu.edu/viewdoc/summary
15. Braga, A.D.P., Ludermir, T.B., Carvalho, A.C.P.D.L.F.: Redes neurais artificiais: teoria e aplicações (2000)
16. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, ??? (2009). https://doi.org/10.1007/978-0-387-21606-5
17. GAMA, J.: Árvores de decisão, 2000. Machine Learning (2009)

18. Garcia, S.C.: O uso de árvores de decisão na descoberta de conhecimento na área da saúde (2003)
19. Noble, W.S.: What is a support vector machine? Nat. Biotechnol. **24**(12), 1565–1567 (2006)
20. Huang, S., Cai, N., Pacheco, P.P., Narrandes, S., Wang, Y., Xu, W.: Applications of support vector machine (svm) learning in cancer genomics. Cancer Genomics Proteomics **15**(1), 41–51 (2018)
21. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. J. Artif. Intell. Res. **11**, 169–198 (1999)
22. Breiman, L.: Bagging predictors. Mach. Learn. **24**, 123–140 (1996)
23. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
24. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
25. LeCun, Y., Bengio, Y., Hinton, G.: others,"deep learning. nature 521 (7553), 436–444," google sch. Google Sch. Cross Ref Cross Ref (2015)
26. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, ??? (2016)
27. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Gated feedback recurrent neural networks. In: International Conference on Machine Learning, pp. 2067–2075 (2015). PMLR
28. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 253–256 (2010). IEEE
29. Bengio, Y., et al.: Learning deep architectures for ai. Foundations trends® in Mach. Learn. **2**(1), 1–127 (2009)
30. Pang, Y., Sun, M., Jiang, X., Li, X.: Convolution in convolution for network in network. IEEE Trans. Neural Netw. Learn. Syst. **29**(5), 1587–1597 (2017)
31. Habibi Aghdam, H., Jahani Heravi, E., Habibi Aghdam, H., Jahani Heravi, E.: Convolutional Neural Networks. Springer (2017)
32. Wu, J.: Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China 5(23), 495 (2017)
33. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing **234**, 11–26 (2017)
34. Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice Hall PTR, ??? (1998)
35. Hochreiter, S.: Untersuchungen zu dynamischen neuronalen netzen [ph. d. dissertation]. Technische Universitt Munchen, Munchen, Germany (1991)
36. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
37. Nelson, D.M.Q.: Uso de redes neurais recorrentes para previsão de séries temporais financeiras (2017)
38. Lu, H., Peng, H., Xu, Z.-D., Matthews, J.C., Wang, N., Iseley, T.: A feature selection–based intelligent framework for predicting maximum depth of corroded pipeline defects. J. Perform. Constr. Facil. **36**(5), 04022044 (2022)
39. Chen, P., Liu, X.: Stress prediction of heated crude oil pipeline in permafrost region via fully coupled heat-moisture-stress numerical simulation and svm algorithm. Tunn. Undergr. Space Technol. **139**, 105210 (2023)
40. Peng, H., Lu, H., Xu, Z.-D., Wang, Y., Zhang, Z.: Predicting solid-particle erosion rate of pipelines using support vector machine with improved sparrow search algorithm. J. Pipeline Syst. Eng. Practice **14**(2), 04022077 (2023)
41. Zhang, M., Ling, J., Tang, B., Dong, S., Zhang, L.: A data-driven based method for pipeline additional stress prediction subject to landslide geohazards. Sustainability **14**(19), 11999 (2022)
42. Li, C., Lan, H.-Q., Sun, Y.-N., Wang, J.-Q.: Detection algorithm of defects on polyethylene gas pipe using image recognition. Int. J. Press. Vessels Pip. **191**, 104381 (2021)
43. Wang, C., Han, F., Zhang, Y., Lu, J.: An sae-based resampling svm ensemble learning paradigm for pipeline leakage detection. Neurocomputing **403**, 237–246 (2020)

44. Jia, Z., Ho, S.-C., Li, Y., Kong, B., Hou, Q.: Multipoint hoop strain measurement based pipeline leakage localization with an optimized support vector regression approach. J. Loss Prev. Process Ind. **62**, 103926 (2019)
45. Wang, W., He, X., Li, Y., Shuai, J.: Risk analysis on corrosion of submarine oil and gas pipelines based on hybrid bayesian network. Ocean Eng. **260**, 111957 (2022)
46. Liu, W., Chen, Z., Hu, Y.: Xgboost algorithm-based prediction of safety assessment for pipelines. Int. J. Press. Vessels Pip. **197**, 104655 (2022)
47. Feng, C., Su, M., Xu, L., Zhao, L., Han, Y., Peng, C.: A novel generalization ability-enhanced approach for corrosion fatigue life prediction of marine welded structures. Int. J. Fatigue **166**, 107222 (2023)
48. Kwong, N.S., Jaiswal, K.S., Baker, J.W., Luco, N., Ludwig, K.A., Stephens, V.J.: Earthquake risk of gas pipelines in the conterminous united states and its sources of uncertainty. ASCE-ASME J. Risk Uncertainty Eng. Syst. Part A Civil Eng. **8**(1), 04021081 (2022)
49. Zhang, W., Ayello, F., Honegger, D., Bozorgnia, Y., Taciroglu, E.: Machine learning-based prediction of the seismic response of fault-crossing natural gas pipelines. Earthq. Eng. Struct. Dynam. **52**(11), 3238–3255 (2023)
50. Chen, Z.-F., Li, X., Sang, Z., Wang, W., Wang, Y.: A novel dynamic parameter method (dpm) based on ann for safety assessment of corroded pipelines. Ocean Eng. **280**, 114922 (2023)
51. Lozovan, V., Skrynkovskyy, R., Yuzevych, V., Yasinskyi, M., Pawlowski, G.: Forming the toolset for development of a system to control quality of operation of underground pipelines by oil and gas enterprises with the use of neural networks. Eastern-Europ. J. Enterp. Technol. **2**(5), 41–48 (2019)
52. Zemenkova, M.Y., Chizhevskaya, E.L., Zemenkov, Y.D.: Intelligent monitoring of the condition of hydrocarbon pipeline transport facilities using neural network technologies. J. Mining Inst. **258**, 933–944 (2022)
53. Li, Z., Liang, Y., Liang, Y., Liao, Q., Wang, B., Huang, L., Zheng, J., Zhang, H.: Review on intelligent pipeline technologies: a life cycle perspective. Comput. Chem. Eng. **175**, 108283 (2023)
54. Gholami, H., Shahrooi, S., Shishehsaz, M.: Strain-based fatigue life analysis of pipelines with external defects under cyclic internal pressure. J. Strain Anal. Eng. Design **56**(5), 313–326 (2021)
55. Shabani, M.M., Shabani, H., Goudarzi, N., Taravati, R.: Probabilistic modelling of free spanning pipelines considering multiple failure modes. Eng. Fail. Anal. **106**, 104169 (2019)
56. Shang, C., Wang, C., Wu, H., Liu, W., Chen, Y., Pan, G., Wang, S., Wu, G., Gao, J., Zhao, H., et al.: Improved data-driven performance of charpy impact toughness via literature-assisted production data in pipeline steel. Sci. China Technol. Sci., 1–11 (2023)
57. Chen, Y., Hou, F., Dong, S., Guo, L., Xia, T., He, G.: Reliability evaluation of corroded pipeline under combined loadings based on back propagation neural network method. Ocean Eng. **262**, 111910 (2022)
58. Li, X., Jing, H., Liu, X., Chen, G., Han, L.: The prediction analysis of failure pressure of pipelines with axial double corrosion defects in cold regions based on the bp neural network. Int. J. Press. Vessels Pip. **202**, 104907 (2023)
59. Xin, J., Chen, J., Li, C., Lu, R.-K., Li, X., Wang, C., Zhu, H., He, R.: Deformation characterization of oil and gas pipeline by acm technique based on ssa-bp neural network model. Measurement **189**, 110654 (2022)
60. Malinowska, A., Cui, X., Salmi, E.F., Hejmanowski, R.: A novel fuzzy approach to gas pipeline risk assessment under influence of ground movement. Int. J. Coal Sci. Technol. **9**(1), 47 (2022)
61. Zhao, B., Li, S., Gao, D., Xu, L., Zhang, Y.: Research on intelligent prediction of hydrogen pipeline leakage fire based on finite ridgelet neural network. Int. J. Hydrogen Energy **47**(55), 23316–23323 (2022)
62. Jiang, S., He, R., Chen, G., Zhu, Y., Shi, J., Liu, K., Chang, Y.: Semi-supervised health assessment of pipeline systems based on optical fiber monitoring. Reliab. Eng. Syst. Saf. **230**, 108932 (2023)

63. Liu, P., Xu, C., Xie, J., Fu, M., Chen, Y., Liu, Z., Zhang, Z.: A cnn-based transfer learning method for leakage detection of pipeline under multiple working conditions with ae signals. Process Saf. Environ. Prot. **170**, 1161–1172 (2023)
64. Li, X., Guo, M., Zhang, R., Chen, G.: A data-driven prediction model for maximum pitting corrosion depth of subsea oil pipelines using ssa-lstm approach. Ocean Eng. **261**, 112062 (2022)
65. Zhou, J., Lin, H., Li, S., Jin, H., Zhao, B., Liu, S.: Leakage diagnosis and localization of the gas extraction pipeline based on sa-pso bp neural network. Reliab. Eng. Syst. Saf. **232**, 109051 (2023)
66. An, J., Liu, P.: Advance prediction method of failure consequence for natural gas pipeline soil corrosion leakage. J. Fail. Anal. Prev. **21**, 2202–2214 (2021)
67. Liang, X., Liang, W., Xiong, J.: Intelligent diagnosis of natural gas pipeline defects using improved flower pollination algorithm and artificial neural network. J. Clean. Prod. **264**, 121655 (2020)
68. Wu, L., Mei, J., Zhao, S.: Pipeline damage identification based on an optimized back-propagation neural network improved by whale optimization algorithm. Appl. Intell. **53**(10), 12937–12954 (2023)
69. Zhang, X., An, J.: A new pre-assessment model for failure-probability-based-planning by neural network. J. Loss Prev. Process Ind. **81**, 104908 (2023)
70. Xie, M., Li, Z., Zhao, J., Pei, X.: A prognostics method based on back propagation neural network for corroded pipelines. Micromachines **12**(12), 1568 (2021)
71. Peng, S., Zhang, Z., Liu, E., Liu, W., Qiao, W.: A new hybrid algorithm model for prediction of internal corrosion rate of multiphase pipeline. J. Nat. Gas Sci. Eng. **85**, 103716 (2021)
72. Ismail, M.F.H., May, Z., Asirvadam, V.S., Nayan, N.A.: Machine-learning-based classification for pipeline corrosion with monte carlo probabilistic analysis. Energies **16**(8), 3589 (2023)
73. Yang, Y., Zheng, P., Zeng, F., Xin, P., He, G., Liao, K.: Metal corrosion rate prediction of small samples using an ensemble
74. Lu, S., Yue, Y., Liu, X., Wu, J., Wang, Y.: A novel unbalanced weighted knn based on svm method for pipeline defect detection using eddy current measurements. Meas. Sci. Technol. **34**(1), 014001 (2022)
75. Zuo, Z., Ma, L., Liang, S., Liang, J., Zhang, H., Liu, T.: A semi-supervised leakage detection method driven by multivariate time series for natural gas gathering pipeline. Process Saf. Environ. Prot. **164**, 468–478 (2022)
76. Liu, J., Su, H., Ma, Y., Wang, G., Wang, Y., Zhang, K.: Chaos characteristics and least squares support vector machines based online pipeline small leakages detection. Chaos, Solitons Fractals **91**, 656–669 (2016)
77. Yin, H., Liu, C., Wu, W., Song, K., Dan, Y., Cheng, G.: An integrated framework for criticality evaluation of oil & gas pipelines based on fuzzy logic inference and machine learning. J. Nat. Gas Sci. Eng. **96**, 104264 (2021)
78. Abyani, M., Bahaari, M.R., Zarrin, M., Nasseri, M.: Predicting failure pressure of the corroded offshore pipelines using an efficient finite element based algorithm and machine learning techniques. Ocean Eng. **254**, 111382 (2022)
79. Mohamed, A., Hamdi, M.S., Tahar, S.: A hybrid intelligent approach for metal-loss defect depth prediction in oil and gas pipelines. In: Intelligent Systems and Applications: Extended and Selected Results from the SAI Intelligent Systems Conference (IntelliSys) 2015. Springer, pp. 1–18 (2016)
80. Lee, L.H., Rajkumar, R., Lo, L.H., Wan, C.H., Isa, D.: Oil and gas pipeline failure prediction system using long range ultrasonic transducers and euclidean-support vector machines classification approach. Expert Syst. Appl. **40**(6), 1925–1934 (2013)
81. Manan, A., Kamal, K., Ratlamwala, T.A.H., Sheikh, M.F., Abro, A.G., Zafar, T.: Failure classification in natural gas pipe-lines using artificial intelligence: a case study. Energy Rep. **7**, 7640–7647 (2021)
82. Naserzadeh, Z., Nohegar, A.: Development of hgapso-svr corrosion prediction approach for offshore oil and gas pipelines. J. Loss Prevent. Process Indus., 105092 (2023)

83. Shaik, N.B., Pedapati, S.R., Taqvi, S.A.A., Ahsan, S., Abd Dzubir, F.A.: Classification of faults in oil and gas pipelines using support vector machines. Pertanika J. Sci. Technol. **28**(S1), 173–184 (2020)
84. Jin, H., Zhang, L., Liang, W., Ding, Q.: Integrated leakage detection and localization model for gas pipelines based on the acoustic wave method. J. Loss Prev. Process Ind. **27**, 74–88 (2014)
85. Du, J., Zheng, J., Liang, Y., Xu, N., Liao, Q., Wang, B., Zhang, H.: Deeppipe: Theory-guided prediction method based automatic machine learning for maximum pitting corrosion depth of oil and gas pipeline. Chem. Eng. Sci., 118927 (2023)
86. Awuku, B., Huang, Y., Yodo, N.: Predicting natural gas pipeline failures caused by natural forces: an artificial intelligence classification approach. Appl. Sci. **13**(7), 4322 (2023)
87. Mazumder, R.K., Salman, A.M., Li, Y.: Failure risk analysis of pipelines using data-driven machine learning algorithms. Struct. Saf. **89**, 102047 (2021)
88. El-Abbasy, M.S., Senouci, A., Zayed, T., Parvizsedghy, L., Mirahadi, F.: Unpiggable oil and gas pipeline condition forecasting models. J. Perform. Constr. Facil. **30**(1), 04014202 (2016)
89. Ossai, C.I.: A data-driven machine learning approach for corrosion risk assessment—a comparative study. Big Data and Cognit. Comput. **3**(2), 28 (2019)
90. Fang, J., Cheng, X., Gai, H., Lin, S., Lou, H.: Development of machine learning algorithms for predicting internal corrosion of crude oil and natural gas pipelines. Comput. Chem. Eng. **177**, 108358 (2023)
91. Seghier, M.E.A.B., H¨oche, D., Zheludkevich, M.: Prediction of the internal corrosion rate for oil and gas pipeline: Implementation of ensemble learning techniques. J. Nat. Gas Sci. Eng. **99**, 104425 (2022)
92. Ahuja, S.K., Shukla, M.K., Ravulakollu, K.K.: Surface corrosion grade classification using convolution neural network. Int. J. Recent Technol. Eng. (IJRTE) (2019)
93. Elmas, F.R., Rios, M.P., Almeida Lima, E.R., Santos, R.S., et al.: Prediction of external corrosion rate in oil and gas platforms using ensemble learning: a maintenance 4.0 approach. Brazilian J. Oper. Prod. Manag. **20**(3), 1952–1952 (2023)
94. Eastvedt, D., Naterer, G., Duan, X.: Detection of faults in subsea pipelines by flow monitoring with regression supervised machine learning. Process Saf. Environ. Prot. **161**, 409–420 (2022)
95. Chen, X., Wang, L., Huang, Z.: Principal component analysis based dynamic fuzzy neural network for internal corrosion rate prediction of gas pipelines. Math. Probl. Eng. **2020**, 1–9 (2020)
96. Quy, T.B., Kim, J.-M.: Real-time leak detection for a gas pipeline using ak-nn classifier and hybrid ae features. Sensors **21**(2), 367 (2021)
97. Li, Y., Sun, C., Liu, Y.: Magnetic flux leakage testing method for pipelines with stress corrosion defects based on improved kernel extreme learning machine. Electronics **12**(17), 3707 (2023)
98. Wang, N., Song, L., Fang, H., Li, B., Wang, F.: Multi-parameter maximum corrosion depth prediction model for buried pipelines based on gscv-xgboost. IEEE Access (2023)
99. Woldesellasse, H., Tesfamariam, S.: Consequence assessment of gas pipeline failure caused by external pitting corrosion using an integrated bayesian belief network and GIS model: application with Alberta pipeline. Reliab. Eng. Syst. Saf. **240**, 109573 (2023)
100. Aalirezaei, A., Kabir, G., Khan, M.S.A.: Dynamic predictive analysis of the consequences of gas pipeline failures using a bayesian network. Int. J. Crit. Infrastruct. Prot. **43**, 100638 (2023)
101. Hong, B., Shao, B., Guo, J., Fu, J., Li, C., Zhu, B.: Dynamic bayesian network risk probability evolution for third-party damage of natural gas pipelines. Appl. Energy **333**, 120620 (2023)
102. Yu, Q., Hou, L., Li, Y., Chai, C., Yang, K., Liu, J.: Pipeline failure assessment based on fuzzy bayesian network and ahp. J. Pipeline Syst. Eng. Pract. **14**(1), 04022059 (2023)
103. Li, X., Jia, R., Zhang, R., Yang, S., Chen, G.: A kpca-brann based data-driven approach to model corrosion degradation of subsea oil pipelines. Reliab. Eng. Syst. Saf. **219**, 108231 (2022)
104. Dao, U., Yarveisy, R., Anwar, S., Khan, F., Zhang, Y., Ngo, H.H.: A bayesian approach to assess under-deposit corrosion in oil and gas pipelines. Process Safety Environ. Protect. (2023)

105. Shabarchin, O., Tesfamariam, S.: Internal corrosion hazard assessment of oil & gas pipelines using bayesian belief network model. J. Loss Prev. Process Ind. **40**, 479–495 (2016)
106. Woldesellasse, H., Tesfamariam, S.: Failure assessment of oil and gas transmission pipelines using an integrated bayesian belief network and gis model. Int. J. Pressure Vessels Piping **205**, 104984 (2023). https://doi.org/10.1016/j.ijpvp.2023.104984
107. Chalgham, W., Wu, K.-Y., Mosleh, A.: System-level prognosis and health monitoring modeling framework and software implementation for gas pipeline system integrity management. J. Nat. Gas Sci. Eng. **84**, 103671 (2020). https://doi.org/10.1016/j.jngse.2020.103671
108. El-Abbasy, M.S., Senouci, A., Zayed, T., Mirahadi, F., Parvizsedghy, L.: Artificial neural network models for predicting condition of offshore oil and gas pipelines. Autom. Constr. **45**, 50–65 (2014)
109. Senouci, A., Elabbasy, M., Elwakil, E., Abdrabou, B., Zayed, T.: A model for predicting failure of oil pipelines. Struct. Infrastruct. Eng. **10**(3), 375–387 (2014)
110. Zakikhani, K., Zayed, T., Abdrabou, B., Senouci, A.: Modeling failure of oil pipelines. J. Perform. Constr. Facil. **34**(1), 04019088 (2020)
111. Wen, K., He, L., Liu, J., Gong, J.: An optimization of artificial neural network modeling methodology for the reliability assessment of corroding natural gas pipelines. J. Loss Prev. Process Ind. **60**, 1–8 (2019)
112. Kumari, P., Halim, S.Z., Kwon, J.S.-I., Quddus, N.: An integrated risk prediction model for corrosion-induced pipeline incidents using artificial neural network and bayesian analysis. Process Saf. Environ. Prot. **167**, 34–44 (2022)
113. Liu, X., Xia, M., Bolati, D., Liu, J., Zheng, Q., Zhang, H.: An ann-based failure pressure prediction method for buried high-strength pipes with stray current corrosion defect. Energy Sci. Eng. **8**(1), 248–259 (2020)
114. Din, M.M., Ithnin, N., Zain, A.M., Noor, N.M., Siraj, M., Rasol, R.: An artificial neural network modeling for pipeline corrosion growth prediction. ARPN J. Eng. Appl. Sci. **10**(2), 512–519 (2015)
115. Muda, M.F., Hashim, M.H.M., Kamarudin, M.K., Mohd, M.H., Rahman, M.A.: Corroded subsea pipelines burst pressure prediction utilizing finite element data using ann. Civil Eng. Archit. **10**(1), 334–344 (2022)
116. Shaik, N.B., Pedapati, S.R., Othman, A., Bingi, K., Dzubir, F.A.A.: An intelligent model to predict the life condition of crude oil pipelines using artificial neural networks. Neural Comput. Appl. **33**(21), 14771–14792 (2021)
117. Lo, M., Karuppanan, S., Ovinis, M.: Failure pressure prediction of a corroded pipeline with longitudinally interacting corrosion defects subjected to combined loadings using fem and ann. J. Marine Sci. Eng. **9**(3), 281 (2021)
118. Wu, Q., Lee, C.-M.: A modified leakage localization method using multilayer perceptron neural networks in a pressurized gas pipe. Appl. Sci. **9**(9), 1954 (2019)
119. Xu, W.-Z., Li, C.B., Choung, J., Lee, J.-M.: Corroded pipeline failure analysis using artificial neural network scheme. Adv. Eng. Softw. **112**, 255–266 (2017)
120. Layouni, M., Hamdi, M.S., Tahar, S.: Detection and sizing of metal-loss defects in oil and gas pipelines using pattern-adapted wavelets and machine learning. Appl. Soft Comput. **52**, 247–261 (2017)
121. Wang, F., Xing, J., Li, J.: Research on the adaptability of an improved high-intelligence long-distance optical fiber pre-warning system. Opt. Eng. **59**(10), 106102 (2020). https://doi.org/10.1117/1.OE.59.10.106102
122. Guo, Y., Meng, X., Wang, D., Meng, T., Liu, S., He, R.: Comprehensive risk evaluation of long-distance oil and gas transportation pipelines using a fuzzy petri net model. J. Natural Gas Sci. Eng. **33**, 18–29 (2016)
123. El-Abbasy, M.S., Senouci, A., Zayed, T., Mirahadi, F., Parvizsedghy, L.: Condition prediction models for oil and gas pipelines using regression analysis. J. Constr. Eng. Manag. **140**(6), 04014013 (2014)
124. Zakikhani, K., Nasiri, F., Zayed, T.: A failure prediction model for corrosion in gas transmission pipelines. Proc. Inst. Mech. Eng. Part O: J. Risk Reliab. **235**(3), 374–390 (2021)

125. Pourahmadi, M., Saybani, M.: Reliability analysis with corrosion defects in sub-marine pipeline case study: oil pipeline in ab-khark island. Ocean Eng. **249**, 110885 (2022)
126. Ramkumar, N., Matharasi, J.A.: Corrosion detection in 'l' bend oil pipelines based on fuzzy implementation. Int. J. Electr. Eng. Technol. **11**, 245–252 (2020)
127. Shang, L., Zhang, Z., Tang, F., Cao, Q., Yodo, N., Pan, H., Lin, Z.: Deep learning enriched automation in damage detection for sustainable operation in pipelines with welding defects under varying embedment conditions. Computation **11**(11), 218 (2023)
128. Spandonidis, C., Theodoropoulos, P., Giannopoulos, F., Galiatsatos, N., Petsa, A.: Evaluation of deep learning approaches for oil & gas pipeline leak detection using wireless sensor networks. Eng. Appl. Artif. Intell. **113**, 104890 (2022)
129. Xiao, B., Miao, S., Xia, D., Huang, H., Zhang, J.: Detecting the backfill pipeline blockage and leakage through an lstm-based deep learning model. Int. J. Miner. Metall. Mater. **30**(8), 1573–1583 (2023)
130. Zhang, H., Zuo, Z., Li, Z., Ma, L., Liang, S.: An unsupervised leak detection method with aggregating prediction and reconstruction along projection pathway for natural gas gathering pipelines. Process Saf. Environ. Prot. **179**, 275–289 (2023)
131. Tian, X., Jiao, W., Liu, T., Ren, L., Song, B.: Intelligent detection method of low-pressure gas system leakage based on semi-supervised anomaly diagnosis. Expert Syst. Appl. **209**, 118376 (2022)
132. Spandonidis, C., Theodoropoulos, P., Giannopoulos, F.: A combined semi-supervised deep learning method for oil leak detection in pipelines using iiot at the edge. Sensors **22**(11), 4105 (2022)
133. Vankov, Y., Rumyantsev, A., Ziganshin, S., Politova, T., Minyazev, R., Zagretdinov, A.: Assessment of the condition of pipelines using convolutional neural networks. Energies **13**(3), 618 (2020)
134. Bastian, B.T., Jaspreeth, N., Ranjith, S.K., Jiji, C.: Visual inspection and characterization of external corrosion in pipelines using deep neural network. NDT E Int. **107**, 102134 (2019)
135. Yang, D., Zhang, X., Zhou, T., Wang, T., Li, J.: A novel pipeline corrosion monitoring method based on piezoelectric active sensing and cnn. Sensors **23**(2), 855 (2023)
136. Zhang, M., Guo, Y., Xie, Q., Zhang, Y., Wang, D., Chen, J.: Estimation of defect size and cross-sectional profile for the oil and gas pipeline using visual deep transfer learning neural network. IEEE Trans. Instrum. Meas. **72**, 1–13 (2022)
137. Wang, Y., Fu, Q., Lin, N., Lan, H., Zhang, H., Ergesh, T.: Identification and classification of defects in pe gas pipelines based on vgg16. Appl. Sci. **12**(22), 11697 (2022)
138. Zhang, M., Guo, Y., Xie, Q., Zhang, Y., Wang, D., Chen, J.: Defect identification for oil and gas pipeline safety based on autonomous deep learning network. Comput. Commun. **195**, 14–26 (2022)
139. Yu, Y., Cheng, X., Wang, L., Wang, C.: Convolutional neural network-based quantitative evaluation for corrosion cracks in oil/gas pipeline by millimeter-wave imaging. IEEE Trans. Instrum. Meas. **71**, 1–9 (2022)
140. Xie, Y., Xiao, Y., Liu, X., Liu, G., Jiang, W., Qin, J.: Time-frequency distribution map-based convolutional neural network (cnn) model for underwater pipeline leakage detection using acoustic signals. Sensors **20**(18), 5040 (2020)
141. Khan, A., Ali, S.S.A., Anwer, A., Adil, S.H., Mériaudeau, F.: Subsea pipeline corrosion estimation by restoring and enhancing degraded underwater images. IEEE Access **6**, 40585–40601 (2018)
142. Xiong, C., Lian, S., Chen, W.: An ensemble method for automatic real-time detection and evaluation of oil and gas leakage in subsea pipelines based on 3d real-time sonar system. J. Civil Struct. Health Monit, 1–19 (2023)
143. Gao, B., Zhao, H., Miao, X.: A novel multi-model cascade framework for pipeline defects detection based on machine vision. Measurement **220**, 113374 (2023)
144. Ma, D., Wang, J., Sun, Q., Hu, X.: A novel broad learning system based leakage detection and universal localization method for pipeline networks. IEEE Access **7**, 42343–42353 (2019). https://doi.org/10.1109/ACCESS.2019.2908015

145. Yao, J., Liang, W., Xiong, J.: Novel intelligent diagnosis method of oil and gas pipeline defects with transfer deep learning and feature fusion. Int. J. Press. Vessels Pip. **200**, 104781 (2022)
146. Miao, X., Zhao, H., Gao, B., Song, F.: Corrosion leakage risk diagnosis of oil and gas pipelines based on semi-supervised domain generalization model. Reliab. Eng. Syst. Saf. **238**, 109486 (2023)
147. Woldesellasse, H., Tesfamariam, S.: Data augmentation using conditional generative adversarial network (cgan): Application for prediction of corrosion pit depth and testing using neural network. J. Pipeline Sci. Eng. **3**(1), 100091 (2023)
148. Akhlaghi, B., Mesghali, H., Ehteshami, M., Mohammadpour, J., Salehi, F., Abbassi, R.: Predictive deep learning for pitting corrosion modeling in buried transmission pipelines. Process Saf. Environ. Prot. **174**, 320–327 (2023)
149. Shaik, N.B., Benjapolakul, W., Pedapati, S.R., Bingi, K., Le, N.T., Asdornwised, W., Chaitusaney, S.: Recurrent neural network-based model for estimating the life condition of a dry gas pipeline. Process Saf. Environ. Prot. **164**, 639–650 (2022)
150. Shaik, N.B., Benjapolakul, W., Pedapati, S.R., Bingi, K., Thien Le, N., Asdornwised, W., Chaitusaney, S.: Recurrent neural network-based model for estimating the life condition of a dry gas pipeline. Process Safety Environ. Protect. **164**, 639–650 (2022) https://doi.org/10.1016/j.psep.2022.06.047
151. Su, Y., Li, J., Yu, B., Zhao, Y., Yao, J.: Fast and accurate prediction of failure pressure of oil and gas defective pipelines using the deep learning model. Reliab. Eng. Syst. Saf. **216**, 108016 (2021)
152. Ferreira, A.D.M., Afonso, S.M., Willmersdorf, R.B., Lyra, P.R.: Multiresolution analysis and deep learning for corroded pipeline failure assessment. Adv. Eng. Softw. **162**, 103066 (2021)
153. Lang, X., Li, P., Cao, J., Li, Y., Ren, H.: A small leak localization method for oil pipelines based on information fusion. IEEE Sens. J. **18**(15), 6115–6122 (2018). https://doi.org/10.1109/JSEN.2018.2840700
154. Obaid, M.H., Hamad, A.H.: Deep learning approach for oil pipeline leakage detection using imagebased edge detection techniques. Journal Européen des Systèmes Automatises **56**(4), 663–673 (2023)

Determination of the Effect of XGBoost's Parameters on a Structural Problem

Yaren Aydın, Sinan Melih Nigdeli, and Gebrail Bekdaş

Abstract Retaining walls are special engineering structures widely used in civil engineering applications for different purposes. Reinforced concrete retaining walls have both structural constraints and constraints such as overturning, shear and soil-bearing capacity. In this chapter, a dataset is generated by optimizing the cantilever-type reinforced concrete retaining wall with Teaching Learning Based Optimization (TLBO). This dataset is analyzed with Extreme Gradient Boosting (XGBoost), one of the machine learning models, and the effect of model parameters on the success of the model is investigated. Performance evaluation was performed using coefficient of determination (R^2). As a result, the "colsample_bytree" parameter had the highest impact on the model.

Keywords Reinforced concrete · TLBO · Optimization · Machine learning · Retaining wall · Parameter effect

1 Introduction

Reasons such as increased human interaction with nature and population growth require the construction of engineering structures. Every engineering structure built meets the needs of people throughout its service life. These structures are exposed to various internal and external impacts. Especially in highway and railway construction phases, steep angle slopes create problems. Retaining walls are used to stabilize these

Y. Aydın · S. M. Nigdeli (✉) · G. Bekdaş
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey
e-mail: melihnig@iuc.edu.tr

Y. Aydın
e-mail: yaren.aydin1@ogr.iuc.edu.tr

G. Bekdaş
e-mail: bekdas@iuc.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_17

steep slopes by preventing them from collapsing, tilting or sliding. Inappropriate design of these retaining structures causes negative economic consequences.

Retaining walls have been widely used in the past. This is supported by simple retaining structures, the Han family in China who drove bamboos into the ground in two rows and filled them with clay to prevent water seepage, and the support of deep trenches with retaining structures since the Middle Ages [1].

The characteristics of retaining structures are to prevent landslides, support the ground and ensure slope stability. are engineering structures built for the purpose. The areas of use of retaining structures are to regulate slopes on roads requiring filling and cutting, to act as edge piers in bridges, to prevent landslides and erosion, to ensure slope stability and to hold deep excavations and structures. When deciding on the type of retaining structure to be used, the height planned to be built, site characteristics, existing building materials, groundwater level, type of soil to be used in filling and preferably cohesion-free soil, suitability to regional conditions and intended use are determined. Types of retaining walls: Weight (massive) retaining walls, semi-weight retaining walls, reinforced concrete retaining walls [2].

In this study, reinforced concrete retaining walls are analyzed among these retaining wall types. Reinforced concrete retaining walls are retaining walls where concrete and steel reinforcement are used together to strengthen the tensile strength of concrete. It is the most widely used type of retaining wall because it can be built higher and reinforced concrete provides more economical solutions. Commonly used types are cantilevered and buttressed reinforced concrete walls. Figure 1 shows a reinforced concrete retaining wall.

As in other engineering structures, cost is very important in retaining walls. Especially in large-scale projects, the wrong type of material selection and the use of inappropriate sections can cause serious financial losses [1].

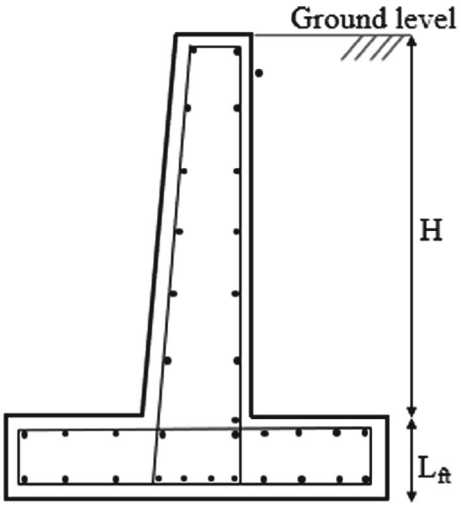


Fig. 1 Reinforced concrete retaining wall

In this chapter, a dataset is created by optimizing a cantilever-type reinforced concrete retaining wall using Teaching Learning Based Optimization (TLBO). Subsequently, this dataset is analyzed using Extreme Gradient Boosting (XGBoost), a machine learning model, to investigate the impact of model parameters on its performance. The evaluation of performance was conducted using the coefficient of determination (R^2).

2 Material and Methods

2.1 Teaching-Learning-Based Optimization (TLBO)

Teaching-learning-based optimization (TLBO) was developed by Rao et al. [3] in 2011. TLBO algorithm is an optimization algorithm that simulates the teaching-learning skill between teacher and student in the classroom.

In the literature, TLBO algorithm is frequently used as a meta-heuristic algorithm technique to solve limited and unlimited non-linear optimization problems. Some of these are reactive power planning [4], photovoltaic cells parameter estimation [5], optimization of a combined heat and power system [6], economic and emission scheduling problems [7] and automatic circle detection [8].

In civil engineering, Toğan [9] carried out the optimum design of planar steel frames using TLBO. Degertekin and Hayalioglu [10] optimized the dimensioning of truss structures using TLBO. Camp and Farshchin [11] have optimally designed space trusses using modified TLBO. Dede [12] used TLBO for weight minimization of truss structures. Bekdaş and Nigdeli [13] used TLBO for the optimum design of reinforced concrete beams. Bekdaş and Nigdeli [14] used TLBO for the optimum design of reinforced concrete columns. Nigdeli and Bekdas [15] adjusted mass dampers to prevent brittle fracture using TLBO. Niğdeli and Bekdaş [16] used TLBO for parameter estimation of double-tuned mass dampers. Kayabekir et al. [17] used TLBO for optimum cost design of reinforced concrete retaining walls. Yücel et al. [18] optimized the dimensions of a reinforced concrete retaining wall using TLBO to minimize the total cost. Aral et al. [19] used TLBO to design the lowest-cost reinforced concrete retaining wall under static and dynamic loads based on reinforced concrete section dimensions and minimization of reinforcement area. Yücel et al. [20] used TLBO for multi-constrained optimum eco-design of reinforced concrete retaining walls. Kundu et al. [21] used TLBO to solve the problem of structural damage detection in frames through changes in vibration responses. Coşut et al. [22] performed cost optimization of a retaining wall under distributed load by using TLBO to satisfy the boundary conditions of the arrangement.

This algorithm is based on a two-stage process: teacher (teacher phase) and interaction with students (student phase). In the algorithm, the teacher is considered to be the individual with the highest level of knowledge in a class that forms the population. Students can have knowledge that can be transferred to them by the teacher. Thus,

the knowledge level of the class increases. The teaching capacity of the teacher and the learning status of the students are interrelated. The two phases of the algorithm, teacher and student, are explained below [3].

Teacher phase is the first stage of the algorithm, the teacher stage, is when teachers transfer knowledge to students. In this stage, a teacher tries to increase the success of the class depending on his/her level of teaching knowledge. The goal of the teacher is to impart knowledge to the students and increase the average knowledge level of the class. The difference between the current average and the new average of the class is given by Eq. 1. TF in Eq. 1 is the teaching factor that decides the value of the mean to be changed and its calculation is given by Eq. 2. The value of TF can be 1 or 2. The value of TF is determined by random probability.

$$Difference\,Mean = rand\,()(X_{i,best} - (TF) \times X_{i,mean}) \tag{1}$$

$$TF = round\,(1 + rand\,()) \tag{2}$$

According to difference mean, the average knowledge level of the students in the teacher phase is updated again in each iteration according to the following expression (Eq. 3).

$$X_{i,new} = X_{i,j} + rand\,()(X_{i,best} - (TF)X_{i,mean}) \tag{3}$$

Here $X_{i,new}$ is the new updated individual. If $X_{i,new}$ is better than the existing result $X_{i,j}$, it replaces $X_{i,j}$ in the next refresh.

After the teacher phase, all best function values are saved for use in the student phase. In this phase, students learn knowledge by interacting and discussing with each other. If one student is more knowledgeable than the other students, the average knowledge level of the other students is updated at each iteration with the help of the knowledgeable student to achieve the desired goal [3]. Random interaction between students improves their knowledge.

For a given learner $X_{i,a}$, another learner $X_{i,b}$ is randomly selected ($a \neq b$). Equation 4 states that if one learner is more knowledgeable than the other learners, the average knowledge level of the other learners is updated at each iteration with the help of the knowledgeable learner to achieve the desired goal.

$$X_{i,new} = \begin{cases} OF_a < OF_b, X_{i,j} + rand\,()(X_{i,a} - X_{i,b}) \\ OF_a > OF_b, X_{i,j} + rand\,()(X_{i,b} - X_{i,a}) \end{cases} \tag{4}$$

Figure 2 shows the TLBO flowchart.

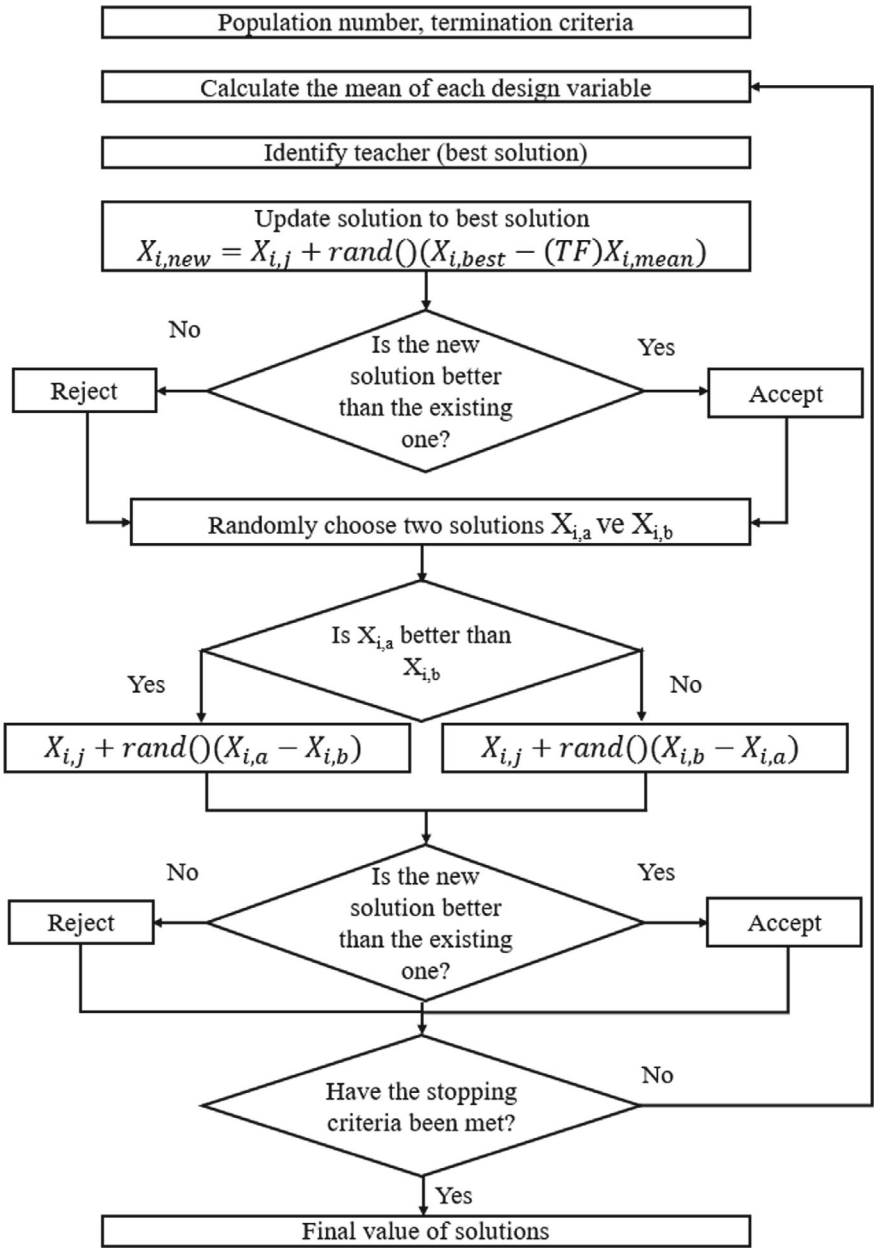


Fig. 2 TLBO flowchart

2.2 Machine Learning (ML)

Machine learning (ML) is a subset of artificial intelligence (AI) that focuses on teaching computers how to make predictions and trying to get a computer to do the kind of things the human mind can do. Although traditional programming can imitate human behavior to a certain extent, when it comes to artificial intelligence, the concepts of machine learning and deep learning often come to mind instead of traditional programming.

Machine learning is widely used in many fields. Many studies have been carried out using machine learning in the field of civil engineering. Keum et al. [23] predicted real-time flood disasters using machine-learning techniques. Karballaeezadeh et al. [24] predicted the pavement structural capacity using machine learning techniques. Bekdaş et al. [25] generated data for the optimum sizing of reinforced concrete circular columns using a harmony search algorithm and predicted them using ensemble learning models. Aydın et al. [26] classified soils using machine learning models such as XGBoost, LightGBM and CatBoost. Aydın et al. [27] used machine learning models for environmentally friendly optimum design of reinforced concrete columns. Aydın et al. [28] used neural network prediction models for carbon emission of alkaline-activated concrete using metaheuristic optimization algorithms. Cakiroglu et al. [29] used machine learning methods such as Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Random Forest (RF) and Categorical Boosting (CatBoost) and SHAP approach to interpretable predictive modeling of tensile strength of basalt fiber reinforced concrete splits. Pham and Nguyen [30] predicted the volume of construction materials using machine learning techniques. Bekdaş et al. [31] performed the optimal sizing of post-tensioned concrete cylindrical walls using SHAP, Harmony Search and ML models such as Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Categorical Gradient Boosting (CatBoost) and Random Forest. Cakiroglu et al. [32] used different ML models to predict the compression capacity of concrete-filled tubular steel (CFST) columns under axial load. Aydın et al. [33] researched hyperparameter optimization in enhancing the accuracy of ANNs for soil type classification.

There are many studies on the design of reinforced concrete retaining walls using optimization and machine learning techniques. Aral et al. [34] used the Jaya algorithm for optimum dimensioning of reinforced concrete (RC) cantilever retaining wall under static loads. Yılmaz et al. [35] used the Jaya algorithm to cost-optimally size a reinforced concrete retaining wall that meets safety conditions under static and dynamic loads. Eroglu et al. [36] optimized the sizing of a reinforced concrete retaining wall under dynamic loads using the Jaya algorithm. Yücel et al. [37] developed a hybrid method using artificial neural networks (ANN) and flower pollination algorithm (FPA) to predict the optimum dimensions of reinforced concrete (RC) cantilever retaining walls. Bekdaş et al. [38] used Random Forest, Extreme Gradient Boosting (XGBoost), Categorical Gradient Boosting (CatBoost) and Light Gradient Boosting Machine (LightGBM) from machine learning models and Harmony Search

(HS) from optimization algorithms for data generation to estimate the optimal dimensions that minimize the construction cost of reinforced concrete retaining walls. Cakiroglu et al. [39] used traditional and ensemble machine-learning techniques for the optimal design of cantilevered soldier pile retaining walls.

In this book chapter, firstly reinforced concrete wall data was generated with the TLBO algorithm and used in the analysis with XGBoost. Thus, the effects of XGBoost parameters on the prediction success of the XGBoost model are investigated.

Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) algorithm is a decision tree-based algorithm developed by Chen and Guestrin in 2016 [40]. The XGBoost algorithm is an optimized version of the Gradient Boosting algorithm with various modifications. The use of XGBoost has become widespread due to its advantages over its predecessors. XGBoost uses the maximum depth value when building the tree, and if the tree shows excessive downward progression, it performs pruning. Thus, excessive learning is prevented. While the Gradient Boosting algorithm uses first-order functions to calculate the loss function, XGBoost performs these calculations using second-order functions. XGBoost's parallel operation feature allows it to reach the result in a shorter time compared to other algorithms [41]. In the XGBoost model, the weights of the classes are determined by creating a tree structure. Trees are created by successively splitting the nodes according to the values of the features with the gradient boosting algorithm. After each tree is trained, the difference between the target and prediction values is calculated. These errors are used to make updates for the next tree. The final prediction value obtained after all trees have been trained is the weighted average of the prediction values of all trees. The prediction based on this weighted average is the final prediction of the XGBoost model. The most important factor in the success of the XGBoost algorithm is that it is 10 times faster than other algorithm solutions and can scale millions of data in its memory [40]. Figure 3 shows the flowchart of XGBoost.

The XGBoost algorithm has multiple hyper parameters. The XGBoost parameters used in this study are given in Table 1.

Descriptions of the parameters in Table 1 are given in the rest of the paper. Gamma is the minimum loss reduction required to make a split. As the value of gamma increases, the algorithm becomes more conservative. The default value of the gamma is 0. Eta (learning rate) controls the step size used in the update to avoid overfitting. The default value of the eta is 0.3. Min_child_weight is the minimum sum of the observation weights required on a child. The algorithm becomes more conservative as its value increases. Colsample_bytree is the subsample rate of columns for each tree. The default value of the colsample_bytree is 1. Subsample is the sub-sample proportion of training samples. Subsampling occurs once in each retrofit iteration. The default value of the subsample is 1. Max_depth, which refers to the maximum depth of a tree, is used to control overfitting. As its value increases, the model becomes more complex and the probability of overfitting increases. The default value of the max_depth is 1 [43].

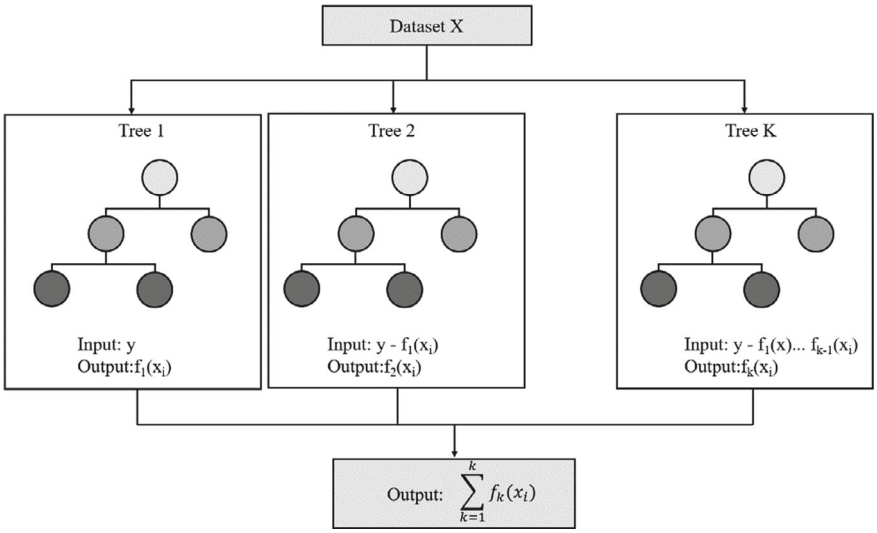


Fig. 3 XGBoost framework [42]

Table 1 XGBoost parameters and value ranges used in the study

Parameter	Value range
"gamma"	[0,∞]
"eta"	[0,1]
"min_child_weight"	[0,∞]
"colsample_bytree"	(0,1]
"subsample"	(0,1]
"max_depth"	[0,∞]

Performance Evaluation

This study uses the coefficient of determination (R^2) for performance evaluation.

The coefficient of determination is a measure of how well the sample regression curve fits the data.

$$R^2 = 1 - \frac{Residual\ Sum\ of\ Squares}{Sum\ of\ All\ Squares} \tag{5}$$

3 Economic Design of RC Wall

Figure 4 shows a cantilever-type reinforced concrete retaining wall and the forces acting on the wall for static loading conditions.

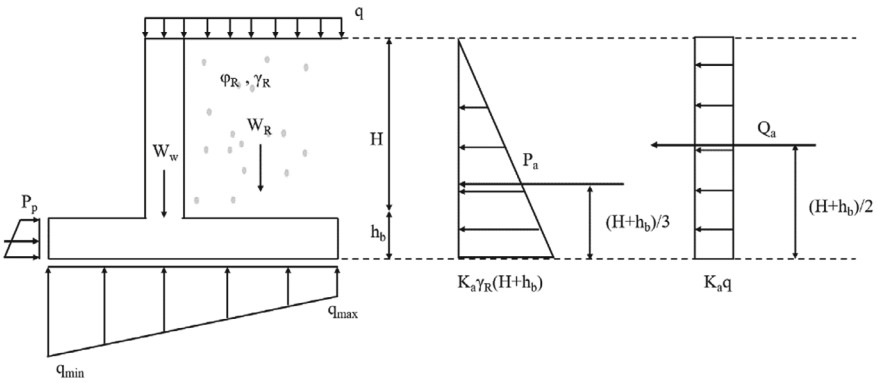


Fig. 4 Forces acting on the wall for static loading condition

The objective of the problem is to design the lowest-cost retaining wall that satisfies all safety conditions for stability and reinforced concrete design. The function defining this objective is given by Eq. 6, which is the sum of the costs of concrete and steel materials used in the retaining wall. In Eq. 6, V_c is the total volume of concrete, C_c is the m^3 unit cost of concrete, W_s is the total weight of steel and C_s is the ton unit cost of steel. Figure 5 shows the cross section and geometry of the optimization problem. The design variables consisting of geometry and reinforcement area related to reinforced concrete design (reinforcement diameters and spacing that can be selected) are given in Table 2.

$$f(x) = C_c V_c + C_s W_s \tag{6}$$

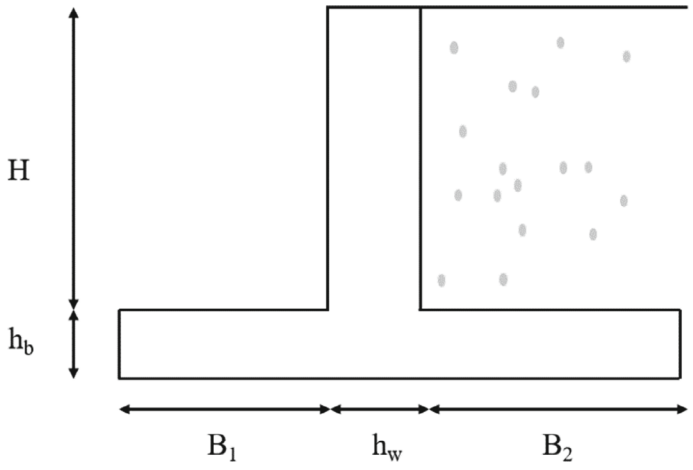


Fig. 5 Geometric design variables of the retaining wall

Table 2 Design variables

	Design variable	Symbol
Variables related to geometry	Heel length	B_2
	Toe length	B_1
	Wall thickness	h_w
	Base plate thickness	h_b
Variables related to reinforced concrete design	Wall (cantilever) reinforcement diameter	R_1
	Wall (cantilever) reinforcement spacing	s_1
	Base plate reinforcement diameter	R_2
	Base plate reinforcement spacing	s_2

The design constants of the problem consist of the unit costs seen in the objective function and various parameters of the material and soil, and these constants are given in Table 3.

The characteristic yield strength of steel (f_{yk}) is 420 MPa and the characteristic compressive strength of concrete (f_{ck}) is 25 MPa. The minimum reinforcement ratio (ρ_{min}) is 0.002 and the spacing (d') is 0.06 m.

The design process of the retaining wall is carried out in two stages: stability checks and reinforced concrete design. The stability checks include three different tests for overturning, sliding and base pressure for the retaining wall. In the first of these, the overturning analysis, the moments of the forces acting on the retaining wall relative to the front end of the base plate are divided into two. The first one is the moments (M_o) which, due to their direction and orientation, will cause the wall to tip over. In Fig. 4, it can be seen that the forces on the back section of the wall are due to the active soil thrust (Eq. 7), denoted by P_A, and the forces on the ground due to the surcharge load (Eq. 8), denoted by Q_A.

Table 3 Design constants

Design constants	Symbol	Value	Unit
Unit volume weight of steel	γ_s	7.85	t/m^3
Unit volume weight of concrete	γ_c	25	kN/m^3
Volume weight of the floor behind the wall	γ_R	18	kN/m^3
Wall (cantilever) height	H	5.6	M
Internal friction angle of the floor behind the wall	φ_R	30	°
Discharge load per unit area	q	10	kPa
Coefficient of friction of the floor under the wall	μ	0.6	–
Bearing capacity of the floor behind the wall	q_u	250	kPa
Unit concrete cost	C_c	120	TL/m^3
Unit steel cost	C_s	1300	TL/ton

$$P_A = 0.5K_a\gamma_R(H + h_b)^2 \tag{7}$$

$$Q_A = K_a q(H + h_b) \tag{8}$$

K_a in the equations is a coefficient related to the soil properties of the active pressure state and is calculated by Eq. 9 depending on the internal friction angle of the soil.

$$K_a = \frac{(cos\varnothing_R)^2}{(1 + sin\varnothing_R)^2} \tag{9}$$

The total overturning effect caused by these forces is given by Eq. 10.

$$\sum M_o = P_A\left(\frac{H + h_b}{3}\right) + Q_A\left(\frac{H + h_b}{2}\right) \tag{10}$$

The second one is the moments (M_R) acting in the opposite direction of the first one, i.e. resisting overturning. In Fig. 4, these forces are the total weight of the retaining wall and the weight of the soil on the rear bulwark, denoted by the symbols WW and WR, respectively. These forces are given in Eqs. 11 and 12 and the total moment generated by the forces is given in Eq. 13. In the calculations performed within the scope of the problem, the passive soil effects (Pp) and the positive effects of the surcharge load (qB_2) are neglected. These effects can be included in this group of forces at the discretion of the designer.

$$W_W = \gamma_c(Hh_w + Bh_b) \tag{11}$$

$$W_R = \gamma_R HB_2 \tag{12}$$

B in Eq. 11 is expressed as the total length of the base plate ($B1 + h_w + B2$).

$$\sum M_R = W_1\left(B_1 + \frac{h_w}{2}\right) + W_2(\frac{B}{2}) + W_R(\frac{B_2}{2} + h_w + B_1) \tag{13}$$

W_1 in Eq. 13 is the weight of the cantilever part of the wall and W_2 is the weight of the base plate.

The safety factor (SF_O), which is used in deciding the wall tensile safety, which is another process of overturning analysis, is the ratio of the sum of moments resisting overturning to the sum of moments causing overturning (Eq. 14) and this coefficient should be greater than the overturning safety coefficient determined in the design.

$$SF_O = \frac{\sum M_R}{\sum M_O} \tag{14}$$

In the second investigation, the slip investigation, there will be forces resisting the retaining wall (F_R) and forces causing the wall to slide (F_D). The sum of the forces resisting slip is expressed by Eq. 15 and the sum of the forces causing slip is expressed by Eq. 16.

$$\sum F_R = \mu(W_W + W_R) \tag{15}$$

$$\sum F_D = P_A + Q_A \tag{16}$$

μ in Eq. 15 is the soil-wall friction coefficient. The factor of safety against sliding (SF_S) coefficient is required to be greater than the shear safety coefficient defined during wall design and is calculated by Eq. 17. In the third and final analysis of the soil stresses under the wall, the stresses due to the axial force and moment created by the forces acting on the wall at the base level are determined. The stresses ($q_{max,min}$) that will occur in a section under the effect of normal force and bending moment are calculated by Eq. 18. A factor of safety (Eq. 19) is calculated so that the smaller of these soil stresses (q_{min}) is greater than zero to avoid tensile stresses in the soil and the larger one (q_{max}) does not exceed the bearing capacity (q_u) of the soil, and this value is aimed to be greater than the soil safety stress coefficient considered during design. At this point, stability checks of the retaining wall were completed. Reinforced concrete design was made according to the cross-sectional effects at critical sections. These critical sections are formed at the base of the cantilever wall of height and the parts of the base plate correspond to the faces of the cantilever wall. Accordingly, the section effects (N_1, V_1, M_1) of the cantilever wall at a critical section can be calculated by Eqs. (20–22) according to Fig. 6.

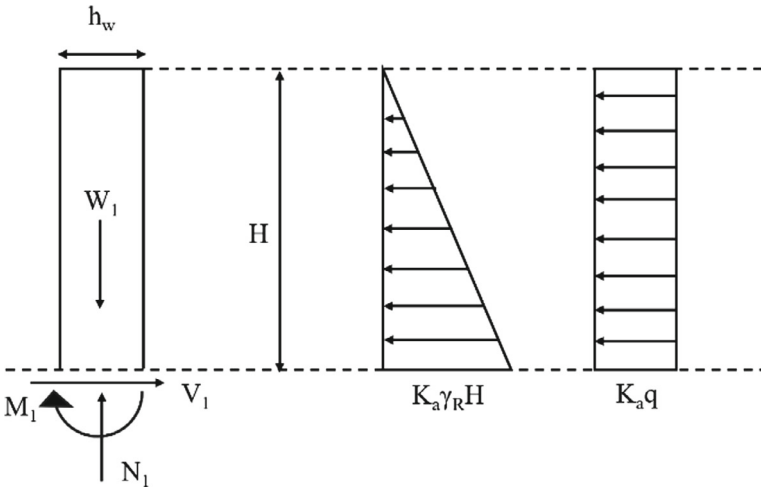


Fig. 6 Forces acting on the cantilever wall and sectional effects at the base of the wall

$$SF_S = \frac{\sum F_R}{\sum F_D} \tag{17}$$

$$q_{max,min} = \frac{N}{A} \pm \frac{M}{W} \tag{18}$$

$$SF_B = \frac{q_u}{q_{max}} \tag{19}$$

$$N_1 = W_1 \tag{20}$$

$$V_1 = 0.5K_a\gamma_R H^2 + K_a qH \tag{21}$$

$$M_1 = \frac{1}{6}K_a\gamma_R H^3 + \frac{1}{2}K_a qH^2 \tag{22}$$

The cross-sectional effects for amplification at the critical section of the base plate are calculated in two different ways according to the moment being positive and negative. If the moment is positive, the cross-sectional effects can be calculated by Eqs. (23–25). The q′ in the equations is the soil stress corresponding to the cantilever wall face and is calculated by Eq. 25. Figure 7 shows the forces acting on the heel and cross-sectional effects on the cantilever wall surface.

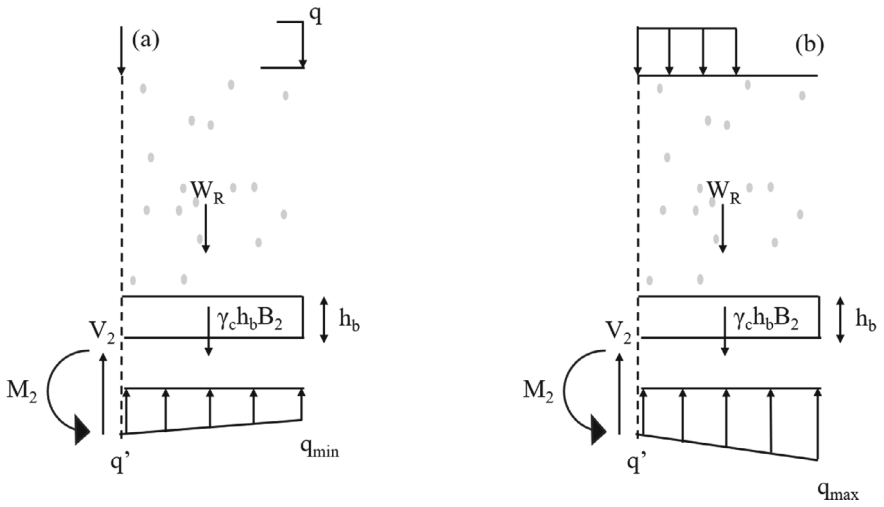


Fig. 7 Forces acting on the heel and cross-sectional effects on the cantilever wall surface

$$V_2 = B_2\left(q + \gamma_c h_b - \frac{q_{min} + q'}{2}\right) + W_R \tag{23}$$

$$M_2 = \frac{{B_2}^2}{2}(q + \gamma_c h_b) + \frac{B_2}{2} - \frac{{B_2}^2}{3}(q_{min} + 0.5q') \tag{24}$$

$$q' = q_{min} + \frac{q_{max} - q_{min}}{B} B_2 \tag{25}$$

Cross-sectional effects can be calculated similarly for the toe. Since there is no dynamic loading on the wall, the reinforced concrete design process is started by defining the calculated sectional effects as design forces. In the reinforced concrete design process, the wall will be calculated only as a member under bending and shear effects due to the relatively small design axial force in the section. Accordingly, Fig. 8 shows the in-section stresses and forces in a reinforced concrete member under the effect of bending moment. Figure 8 shows the rectangular section element under the effect of bending moment.

According to the moment equilibrium condition, the bending moment acting on the section is equal to the moment value formed by the compressive and tensile forces (Eq. 26).

$$0.85 f_{cd} ab = A_s f_{yd} \tag{26}$$

The depth of the pressure block (a) in Fig. 8 is calculated by Eq. 27 and the required reinforcement area (A_s) is calculated by Eq. 28. In the equations, f_{cd} is the design compressive strength of concrete and f_{yd} is the design yield strength of steel.

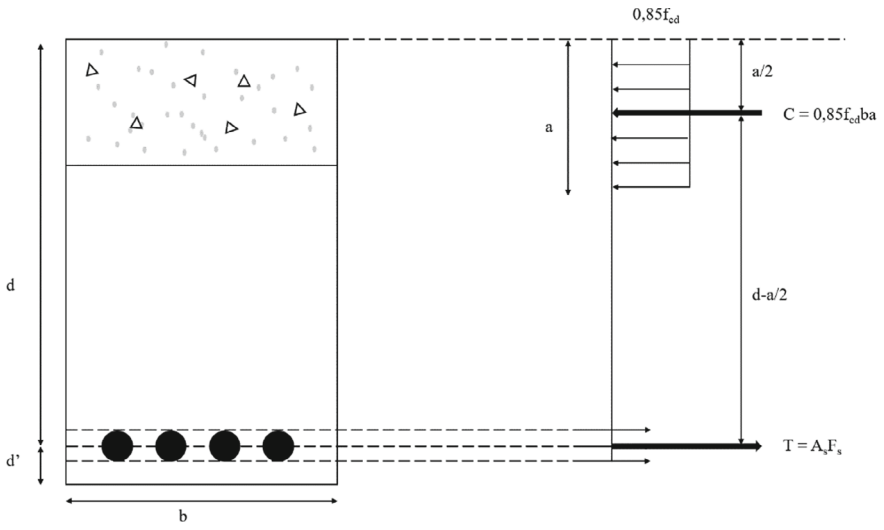


Fig. 8 Rectangular section element under the effect of bending moment

Table 4 Design constraints

	Description	Design constraints
Stability condition	Rollover safety	$SF_{o,design} > SF_o$
	Slip safety	$SF_{S,design} > SF_S$
	Safety of soil safety stress	$SF_{B,design} > SF_B$
	Maximum bending capacity safety (q_{min})	$q_{min} \geq 0$
Reinforced concrete design	Bending capacity at critical sections (M_{cr})	$M_{cr} \geq M_u$
	Shear force capacity at critical sections (V_{cr})	$V_{cr} \geq V_u$
	Minimum reinforcement ratio at critical sections (As_{min})	$As \geq As_{min}$
	Maximum reinforcement spacing at critical sections (S_{max})	$S \leq S_{max}$
	Mimimum concrete cover (d′)	$d' \geq 40$ mm

$$a = d - \sqrt{d^2 - \frac{2M_u}{0.85 f_{cd} b}} \tag{27}$$

$$A_s = \frac{0.85 f_{cd} ab}{f_{yd}} \tag{28}$$

In the final stage of the design, checks are made according to the shear force. This shear capacity is calculated by Eq. 29 based on the shear capacity of the concrete section only. V_{cr} in the equation is the critical shear force that will cause the section to crack.

$$V_{cr} = 0.65 f_{ctd} bd \tag{29}$$

The above-mentioned safety conditions in stability checks and reinforced concrete design and the conditions specified in the regulation (TS500) constitute the constraints of the problem. The design constraints to be applied within the scope of this problem are summarized in Table 4.

4 Results

6 different XGBRegressor parameters were used in the study and their effects on the success of the model are given in 3 tables. In Table 5, the effects of " gamma" and "eta" parameters are analysed. When the " gamma" parameter is analysed in Table 5, slightly the same R^2 values are obtained in 4 different values. When the "eta" parameter is analysed, at a value of 1, it is observed that the R^2 value decreases.

Table 5 Effects of the gamma and eta parameters

Parameters	Model architecture	R^2
Default	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9728366926390506
gamma	gamma = "0.8", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9734942301805387
	gamma = "2", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9733926491768162
	gamma = "10", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.972374658361528
eta	gamma = "0", eta = "0.1", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9732329880380652
	gamma = "0", eta = "0.7", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.973045172412496
	gamma = "0", eta = "1", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9521287877046039

In Table 6 the effects of "min_child_weight" and "colsample_bytree" parameters are analysed. When the "min_child_weight" parameter is analysed in Table 6, a value of 50 decreases the R^2. When the "colsample_bytree" parameter is analysed, it is seen that as the value of "colsample_bytree" increases, the success of the model increases.

In Table 7 the effects of "subsample" and "max_depth" parameters are analysed. When the "min_samples_leaf" parameter is analysed in Table 7, it is seen that as the value of "subsample" increases, the success of the model increases. When the "max_depth" parameter is analysed from Table 7, among the 3 different values tried, the highest success was achieved with a value of 10.

Table 6 Effects of the min_child_weight and colsample_bytree parameters

Parameters	Model architecture	R^2
Default	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9728366926390506
min_child_weight	gamma = "", eta = "0.3", min_child_weight = "0", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9728366926390506
	gamma = "0", eta = "0.3", min_child_weight = "10", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9790012554508484
	gamma = "0", eta = "0.3", min_child_weight = "50", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9329144890109422
colsample_bytree	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "0.4", subsample = "1" and max_depth = "6"	0.7795365617602359
	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "0.6", subsample = "1" and max_depth = "6"	0.7795365617602359
	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "0.8", subsample = "1" and max_depth = "6"	0.954067027582636
	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9728366926390506

Table 7 Effects of the subsample and max_depth parameters

Parameters	Model architecture	R^2
Default	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "6"	0.9728366926390506
subsample	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "0.3" and max_depth = "6"	0.9710668997365413
	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "0.6" and max_depth = "6"	0.9831949857182707
	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "0.9" and max_depth = "6"	0.9820974785236218
max_depth	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "0"	0.9692125814407115
	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "10"	0.9764882022709241
	gamma = "0", eta = "0.3", min_child_weight = "1", colsample_bytree = "1", subsample = "1" and max_depth = "100"	0.9756199767912707

5 Conclusion

Optimization and machine learning are methods that are frequently used to solve problems today. In this study, firstly, the cost optimization of the cantilever-type reinforced concrete retaining wall is performed with the optimisation process. The Extreme Gradient Boosting Regressor (XGBoostRegressor) model is used to predict the cost of the cantilever-type reinforced concrete retaining wall on the 600-row dataset obtained as a result of the optimization and the effect of XGBoost parameters on performance is examined. The coefficient of determination (R^2) was used in performance evaluation.

The parameters whose effects on the model are examined in the study are "gamma", "eta", "min_child_weight", "colsample_bytree", "subsample" and "max_depth".

Among the parameters used in the study, the "colsample_bytree" parameter had the highest effect. Values of 0.4 and 0.6 of this parameter decreased the R^2 value of the default model from 0.97 to 0.77. A small value of the colsample_bytree parameter, which defines what percentage of the features are used to build each tree, means that features are underutilized and caused the largest change in the R^2 value. Changing the values of the other parameters did not cause a significant difference in the model's performance. This study demonstrates the success of parameter variation on the

model through the example of minimum cost of a cantilever type reinforced concrete retaining wall.

References

1. Önalp, A.: Geotechnical knowledge for civil engineers. Karadeniz Technical University, Trabzon (1982)
2. TS 7994.: Soil Retaining Structures, Classification Properties and Project Design Principles, Turkish Standards Institute, Ankara (1990)
3. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. **43**(3), 303–315 (2011). https://doi.org/10.1016/j.cad.2010.12.015
4. Bhattacharyya, B., Babu, R.: Teaching learning based optimization algorithm for reactive power planning. Int. J. Electr. Power Energy Syst. **81**, 248–253 (2016). https://doi.org/10.1016/j.ijepes.2016.02.042
5. Ramadan, A., Kamel, S., Korashy, A., et al.: Photovoltaic cells parameter estimation using an enhanced teaching–learning-based optimization algorithm. Iran J. Sci. Technol. Trans. Electr. Eng. **44**, 767–779 (2020). https://doi.org/10.1007/s40998-019-00257-9
6. Toopshekan, A., Abedian, A., Azizi, A., Ahmadi, E., Rad, M.A.V.: Optimization of a CHP system using a forecasting dispatch and teaching-learning-based optimization algorithm. Energy **285**, 128671 (2023). https://doi.org/10.1016/j.energy.2023.128671
7. Rani, S., Roy, S., Bhattacharjee, K., Bhattacharya, A.: Teaching learning based optimization to solve economic and emission scheduling problems. In: 2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC), pp. 546–550. IEEE (2016)
8. Lopez-Martinez, A., Cuevas, F.J.: Automatic circle detection on images using the teaching learning based optimization algorithm and gradient analysis. Appl. Intell. **49**, 2001–2016 (2019). https://doi.org/10.1007/s10489-018-1372-2
9. Toğan, V.: Design of planar steel frames using teaching–learning based optimization. Eng. Struct. **34**, 225–232 (2012). https://doi.org/10.1016/j.engstruct.2011.08.035
10. Degertekin, S.O., Hayalioglu, M.S.: Sizing truss structures using teaching-learning-based optimization. Comput. Struct. **119**, 177–188 (2013). https://doi.org/10.1016/j.compstruc.2012.12.011
11. Camp, C.V., Farshchin, M.: Design of space trusses using modified teaching–learning based optimization. Eng. Struct. **62**, 87–97 (2014). https://doi.org/10.1016/j.engstruct.2014.01.020
12. Dede, T.: Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures. KSCE J. Civ. Eng. **18**, 1759–1767 (2014). https://doi.org/10.1007/s12205-014-0553-8
13. Bekdaş, G., Nigdeli, S.M.: Optimum design of reinforced concrete beams using teaching-learning-based optimization. In: 3rd International Conference on Optimization Techniques in Engineering (OTENG'15), pp. 7–9 (2015)
14. Bekdaş, G., Nigdeli, S.M.: Optimum design of reinforced concrete columns employing teaching learning based optimization. Chall. J. Struct. Mech. **2**(4), 216–219 (2016). https://doi.org/10.20528/cjsmec.2016.11.030
15. Nigdeli, S.M., Bekdas, G.: Tuning of mass dampers for preventing brittle fracture by employing teaching learning based optimization. PAMM **16**(1), 709–710 (2016). https://doi.org/10.1002/pamm.201610343
16. Niğdeli, S.M., Bekdaş, G.: Teaching-learning based optimization for parameter estimation of double tuned mass dampers. Challenge **3**(2), 90–95 (2017). https://doi.org/10.20528/cjsmec.2016.11.032
17. Kayabekir, A.E., Yücel, M., Bekdaş, G., Nigdeli, S.M.: Comparative study of optimum cost design of reinforced concrete retaining wall via metaheuristics. Chall. J. Concr. Res. Lett. **11**, 75–81 (2020). https://doi.org/10.20528/cjcrl.2020.03.004

18. Yücel, M., Kayabekir, A.E., Bekdaş, G., Nigdeli, S.M., Kim, S., Geem, Z.W.: Adaptive-hybrid harmony search algorithm for multi-constrained optimum eco-design of reinforced concrete retaining walls. Sustainability **13**(4), 1639 (2021). https://doi.org/10.3390/su13041639
19. Aral, S., Yılmaz, N., Bekdaş, G., Nigdeli, S.M.: Jaya optimization for the design of cantilever retaining walls with toe projection restriction. In: Proceedings of 6th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2020, Istanbul, pp. 197–206. Springer Singapore (2021). https://doi.org/10.1007/978-981-15-8603-3_18
20. Yücel, M., Bekdaş, G., Nigdeli, S.M., Kayabekir, A.E.: Optimum design of reinforced concrete retaining walls by using specific parameter-free metaheuristic algorithms. In: Proceedings of the 14th ECCOMAS ematic Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control (2021). https://doi.org/10.7712/140121.7945.18379
21. Kundu, R.D., Mishra, M., Maity, D.: Teaching–learning-based optimization algorithm for solving structural damage detection problem in frames via changes in vibration responses. Arch., Struct. Constr. **3**(4), 395–414 (2023). https://doi.org/10.1007/s44150-021-00009-6
22. Çoşut, M., Bekdaş, G., Nigdeli, S.M.: The cost optimization of reinforced concrete retaining wall with different algorithms. In: International Conference on Intelligent Computing & Optimization, pp. 289–298. Cham: Springer Nature Switzerland (2023). https://doi.org/10.1007/978-3-031-50151-7_28
23. Keum, H.J., Han, K.Y., Kim, H.I.: Real-time flood disaster prediction system by applying machine learning technique. KSCE J. Civ. Eng. **24**(9), 2835–2848 (2020). https://doi.org/10.1007/s12205-020-1677-7
24. Karballaeezadeh, N., Ghasemzadeh Tehrani, H., Mohammadzadeh Shadmehri, D., Shamshirband, S.: Estimation of flexible pavement structural capacity using machine learning techniques. Front. Struct. Civ. Eng. **14**, 1083–1096 (2020). https://doi.org/10.1007/s11709-020-0654-z
25. Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z.W.: Optimization and predictive modeling of reinforced concrete circular columns. Materials **15**(19), 6624 (2022). https://doi.org/10.3390/ma15196624
26. Aydın, Y., Işıkdağ, Ü., Bekdaş, G., Nigdeli, S.M., Geem, Z.W.: Use of machine learning techniques in soil classification. Sustainability **15**(3), 2374 (2023). https://doi.org/10.3390/su15032374
27. Aydın, Y., Bekdaş, G., Nigdeli, S.M., Isıkdağ, Ü., Kim, S., Geem, Z.W.: Machine learning models for ecofriendly optimum design of reinforced concrete columns. Appl. Sci. **13**(7), 4117 (2023). https://doi.org/10.3390/app13074117
28. Aydın, Y., Cakiroglu, C., Bekdaş, G., Işıkdağ, Ü., Kim, S., Hong, J., Geem, Z.W.: Neural network predictive models for alkali-activated concrete carbon emission using metaheuristic optimization algorithms. Sustainability **16**(1), 142 (2023). https://doi.org/10.3390/su16010142
29. Cakiroglu, C., Aydın, Y., Bekdaş, G., Geem, Z.W.: Interpretable predictive modelling of basalt fiber reinforced concrete splitting tensile strength using ensemble machine learning methods and SHAP approach. Materials **16**(13), 4578 (2023). https://doi.org/10.3390/ma16134578
30. Pham, V.H.S., Nguyen, N.L.: Estimating the volume of civil construction materials by machine learning models. Asian J. Civ. Eng. **24**(7), 2077–2088 (2023). https://doi.org/10.1007/s42107-023-00625-1
31. Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z.W.: Optimal dimensions of post-tensioned concrete cylindrical walls using harmony search and ensemble learning with SHAP. Sustainability **15**(10), 7890 (2023). https://doi.org/10.3390/su15107890
32. Cakiroglu, C., Islam, K., Bekdaş, G., Isikdag, U., Mangalathu, S.: Explainable machine learning models for predicting the axial compression capacity of concrete filled steel tubular columns. Constr. Build. Mater. **356**, 129227 (2022). https://doi.org/10.1016/j.conbuildmat.2022.129227
33. Aydın, Y., Bekdaş, G., Isikdag, U., Nigdeli, S.M., Geem, Z.W.: Optimizing artificial neural network architectures for enhanced soil type classification. Geomech. Eng. Int. J. **37**(3), 263–277 (2024). https://doi.org/10.12989/gae.2024.37.3.263
34. Aral, S., Bekdaş, G., Nigdeli, S.M.: Comparative study on optimization of cantilever retaining walls via several metaheuristics. In: Proceedings of 7th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2022, pp. 63–71. Singapore: Springer Nature Singapore (2022). https://doi.org/10.1007/978-981-19-2948-9_7

35. Yılmaz, N., Aral, S., Nigdeli, S.M., Bekdaş, G.: Optimum design of reinforced concrete retaining walls under static and dynamic loads using Jaya Algorithm. In: Proceedings of 6th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2020, Istanbul, pp. 187–196. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-8603-3_17
36. Eroglu, N., Aral, S., Nigdeli, S.M., Bekdas, G.: Jaya algorithm based optimum design of reinforced concrete retaining walls under dynamic loads. Chall. J. Struct. Mech. **7**(2), 64–70 (2021). https://doi.org/10.20528/cjsmec.2021.02.002
37. Yücel, M., Bekdaş, G., Nigdeli, S.M., Kayabekir, A.E.: An artificial intelligence-based prediction model for optimum design variables of reinforced concrete retaining walls. Int. J. Geomech. **21**(12), 04021244 (2021). https://doi.org/10.1061/(ASCE)GM.1943-5622.0002234
38. Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z.W.: Optimal dimensioning of retaining walls using explainable ensemble learning algorithms. Materials **15**(14), 4993 (2022). https://doi.org/10.3390/ma15144993
39. Cakiroglu, C., Islam, K., Bekdaş, G., Nehdi, M.L.: Data-driven ensemble learning approach for optimal design of cantilever soldier pile retaining walls. In: Structures, vol. 51, pp. 1268–1280. Elsevier (2023). https://doi.org/10.1016/j.istruc.2023.03.109
40. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD İnternational Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016). https://doi.org/10.1145/2939672.2939785
41. Kelle, A.C., Yüce, H.: Classification of DoS attacks in MQTT network with machine learning and ınterpretation of the model with SHAP. J. Mater. Mechatron.: A **3**(1), 50–62 (2022). https://doi.org/10.55546/jmm.995091
42. Liu, J.J., Liu, J.C.: Permeability predictions for tight sandstone reservoir using explainable machine learning and particle swarm optimization. Geofluids **2022**, 1–15 (2022). https://doi.org/10.1155/2022/2263329
43. XGBoost Parameters, https://xgboost.readthedocs.io/en/stable/parameter.html

Area Optimization of Bending Members with Different Shapes in Terms of Pure Bending

Muhammed Çoşut, Gebrail Bekdaş, and Sinan Melih Nigdeli

Abstract This study focuses on the optimization of the total cross-sectional area of bending elements with different shapes under the influence of loads. The optimization process is conducted using the Grey Wolf Optimization (GWO), which is a metaheuristic algorithm implemented in the Matlab program. Bending elements offer a wide range of usage options, making simple bending analysis highly significant and applicable in various areas. Therefore, performing optimization studies and achieving designs based on structural constraints are of great importance, aiming to control material utilization at a defined level (minimizing raw material consumption), attain cross-sectional shapes with optimum areas as well as reduce costs. To determine the cross-section of elements with two distinct shapes, the algorithm assigns variables for section lengths and heights, subsequently locating the centre of gravity of the system. Following this step, the distances from the centres of each component to the axes are calculated, enabling the computation of moment of inertia for both directions. Once the moments of inertia are obtained, the bending moments and stresses are computed. Considering that the design of each section is desired based on a specific stress value, constraint checks are conducted, and the decision regarding whether to penalize or not penalize the objective function is determined, ultimately leading to achieving the minimum area for each shape. Notably, due to the variation in shapes, it is observed that each element possesses a different minimum area.

Keywords Pure bending · Section design · Metaheuristic algorithm · Grey Wolf Optimization · Cross-section area optimization

M. Çoşut · G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Avcılar, İstanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_18

1 Introduction

Numerous studies have been conducted and attempted to be implemented in various fields on the subject of bending. Within the scope of these studies, disciplines such as civil engineering, mechanical engineering, and related engineering fields carry out their investigations based on the attainment of necessary and sufficient capacity. In the context of bending analysis, pure bending is utilized to analyze and control displacements, bending stresses, and deformation responses of systems, structures, or beams under loads (normal loads or moments). It ensures that the stress and deformation values of the design elements are maintained within specified levels, thus preventing the system from suffering damage. Additionally, systems or structural components can be designed according to specific requirements. In such cases, the design is completed by considering design criteria and boundary conditions, followed by conducting analyses and checks under loading conditions. These analyses are based on the assumption that the element behaves within the elastic limits. Recently, the analysis and design of systems with lower costs or lower CO_2 emissions have gained significant attention by utilizing algorithms. In general, metaheuristic algorithms inspired by nature are employed, and these exhibit variations in terms of the number of phases and equations involved, which are crucial factors influencing the convergence speed.

Hill [1], Timoshenko and Goodier [2] and Alexander [3] have studied to develop pure bending to calculate element stress and strain along with deflection. Elchalakani et al. [4] took into consideration the behaviour of empty void-filled, cold-formed circular hollow sections in terms of pure plastic bending in order to reach section capacity. Zibasokhan et al. [5] have studied braced structures in order to predict behaviour of the proposed dissipater pertaining the pure bending yielding. For complex C-sections, Pham and Hancock [6] have investigated the strength design of high strength and inelastic bucking strength design in pure bending. Magnucki and Paczos [7] have optimized the cold-formed thin-walled channel beams with flanges in pure bending as theoretical shape, and the objective function was defined as dimensionless. Naik et al. [8] have operated the minimum stress and weight reduction of hole shapes in beams. Qiao and Li [9] have studied pure bending beam optimization, and the objective functions are generated as the minimization of material, stress as well as system displacement.

When beams are thoroughly analysed, it is evident that they not only transmit the loads in the structure to the columns but also play a crucial role in absorbing and dissipating the loads on the supporting elements under dynamic loading conditions. The applied loads on the beam induce bending effects, which are more significant than rotational, axial, or compressive effects, making bending a critical factor to be considered for beam design [10]. Thus, the design of beams primarily revolves around addressing the bending condition.

In this study, the area optimization of the cross-section for bending elements with different shapes was carried out in case of keeping at a certain level under loads. Thus, by finding the moment of inertia and centre of gravity of the system, the beams

with the minimum area were designed according to the stress values, and these cross-section dimensions were found by Grey Wolf Optimization (GWO), which is a metaheuristic algorithm.

2 Pure Bending

Beams are commonly used structural elements in buildings. According to regulatory conditions, beams are subject to certain limitations, and in seismic regions, different structural elements need to be used in conjunction with beams for specific building heights [11]. Therefore, the use of beams in structural applications is primarily focused on load transfer and connection of load-bearing elements [12]. Load transfer involves the transmission of dead and live loads from slabs to beams and further to columns, while providing the connection between columns and shear walls to transfer dynamic loads along with the displacement of the structure from one vertical load-bearing element to another.

Figure 1 illustrates the forces acting on a beam with the neutral axis (x-axis) shown. These forces are distributed in tension or compression depending on the direction of the applied forces or moments. In the case of a moment acting in a clockwise direction, compression bending stresses occur in the upper part of the neutral axis, while tension bending stresses occur in the down part of the neutral axis. Conversely, when a moment acts in the opposite direction, tension and compression bending stresses occur in the upper and lower regions, respectively.

Figure 2 demonstrates the bending behaviour experienced by a specific cross-section under the influence of a moment. Depending on the direction of the applied moment, a bending deformation is observed, resulting in tensile and compressive stresses within the system. In regions where tensile stresses occur, the distances between fibres increase, leading to a certain degree of elongation. Conversely, in regions subjected to compressive stresses, the distances between fibres decrease,

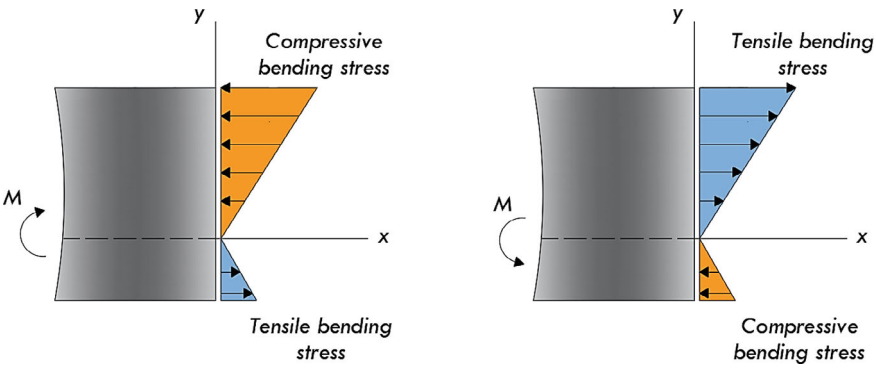


Fig. 1 Compressive and tensile zone

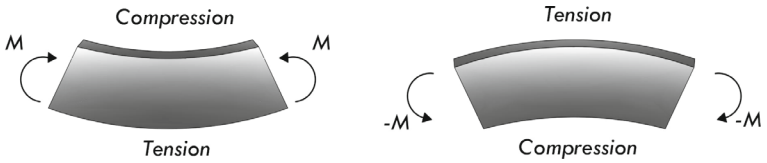


Fig. 2 Compression and tension according to moment direction

causing shrinkage. These effects necessitate the careful arrangement of designs to accommodate such phenomena. In regions experiencing tensile stresses, the utilization of reinforcing elements capable of withstanding tension is crucial to prevent the structure from cracking and sustaining damage. Furthermore, in some cases, additional reinforcement is incorporated in compressive sides to ensure the successful functioning of the system, particularly under high load magnitudes.

The determination of stresses generated in elements with diverse geometries under applied loads entails a systematic approach consisting of several stages. These stages are typically executed in the following prescribed order.

– The centre of gravity (Eq. 1) of each part in the element is found separately according to the x-axis and the y-axis. Figure 3 delineates the centre of gravity and area for the all part of the element.

$$G_y = \frac{A_1 \times y_1 + A_2 \times y_2}{A_1 + A_2} \tag{1}$$

There are several types of shapes; thus, the equations should organise according to the shapes which are used in the system for the assessment of the criteria. As for Fig. 4, G_3 and A_3 should be added to Eq. 1, so it provides to find the center of the shapes.

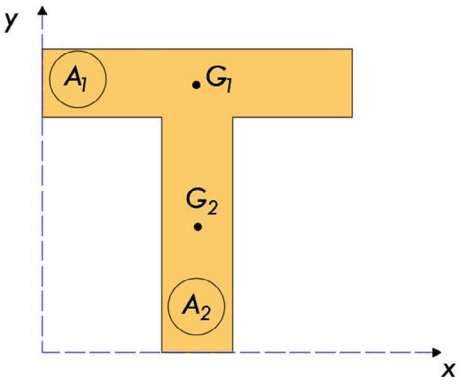


Fig. 3 T shape for the computing

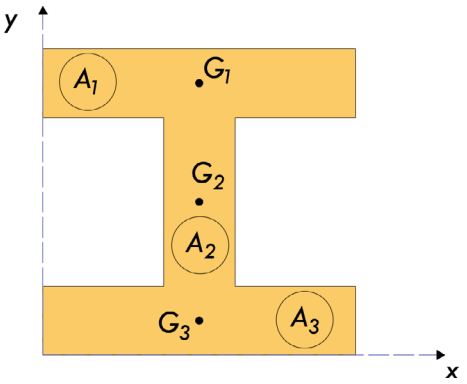


Fig. 4 I shape for computing

– Moments of inertia are found for the x and y axis, and the value of moment of inertia which should be exploited in the problems is also calculated at this stage (Eqs. 2 and 3).

$$I_x = \sum_{n:1}^{n} \left(\frac{b \times h^3}{12} + A \times y^2 \right) \tag{2}$$

$$I_y = \sum_{n:1}^{n} \left(\frac{h \times b^3}{12} + A \times y^2 \right) \tag{3}$$

– The product moment of inertia is computed in Eq. 4. If the product moment of inertia is equal to zero, the XY axis set is accepted as the principal axis set and the axis is found as the neutral axis.

$$I_{xy} = \sum_{n:1}^{n} (A \times x \times y) \tag{4}$$

– According to the centre of gravity of the section and the loading, section modulus calculations are performed in lower and upper regions (Eqs. 5 as well as 6).

$$W_{bottom} = \frac{I_y}{h_{bottom}} \tag{5}$$

$$W_{top} = \frac{I_y}{h_{top}} \tag{6}$$

– After obtaining the aforementioned values, maximum and minimum stress values occurring in the section are computed, and their distribution of stress is depicted

on the shape. M_y represents the moment value in the section.

$$\sigma_{bottom} = \frac{M_y}{W_{bottom}} \tag{7}$$

$$\sigma_{top} = \frac{M_y}{W_{top}} \tag{8}$$

3 Metaheuristic Algorithm

Metaheuristic algorithms which have proven to be beneficial in optimizing various aspects such as structural design, transportation planning, and resource management are developed by drawing inspiration from natural phenomena (dominantly) [13] as well as social life and abstract concepts. These algorithms provide engineers with powerful tools [14] to find optimal solutions that meet specific criteria and constraints while considering factors such as structural integrity, sustainability, and environmental impact. Moreover, the main properties of any metaheuristic algorithms are diversification, exploitation along with intensification [15–19]. The objective functions can be determined according to the uniqueness of the problem, and algorithms are designed accordingly to achieve the objective function. These objective functions can include cost, displacement of the system, reduction of CO_2 emissions, or optimization of system weight. Compared to systems designed using traditional methods, these algorithms are widely used to achieve efficient [20, 21] resource utilization and reduce CO_2 emissions.

To optimize the desired problem, the fixed values, constraints, and variables are determined according to standards or specific conditions and incorporated into the algorithms. Equations are added depending on the type of algorithm to be used in optimization, and each algorithm has its own set of equations and phase numbers. The existence of these differences leads to convergence and achievement of the objective function at different iteration values for different problems. When determining the objective function, problems with relatively few variables and solutions do not exhibit significant differences in iteration count and time. However, complex problems with a large number of variables can exhibit significant differences. Therefore, hybrid algorithms have been developed by combining two or more metaheuristic algorithms to facilitate the more effective solution to complex problems.

3.1 Grey Wolf Optimization

Grey Wolf Optimization which was developed taking into account their leadership and their hunting nature was found by Mirjalili et al. [22] in 2014 [23].

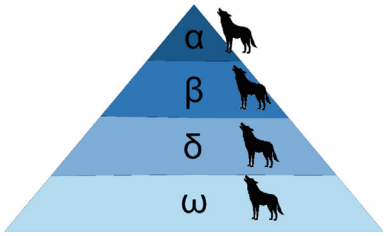


Fig. 5 Grey wolf's hierarchy (dominance increases from down to top)

Grey wolves generally live as 5–12 individuals in nature. It is known that during this period, each individual has a different status and is divided into groups as alpha, beta, delta and omega. These name differences were created according to the situation where wolves have different duties in the herd and these duties are named according to their content. As the leader of the group, the alpha wolf provides the hierarchy and has responsibilities such as hunting and determining vital needs such as shelter. The selection of alpha wolf is chosen to be someone who can lead the group, rather than being the strongest individual in the group. Beta wolf is in the second step of leadership. They convey the orders he received from the alpha wolf to the delta and omega wolves and asks for it to be done. Additionally, if the alpha wolf dies, the beta wolf has the potential to become the leader of the group. In the third and fourth steps of the pyramid, there are delta and omega wolves, respectively. The delta wolf watches the group's borders and warns against future danger and it provides security. They also assist alphas and betas in hunting situations and protect the wounded wolf in case of injury. The omega wolf has no superiority over any other wolf in the pyramid (Fig. 5).

The situation which is within the scope of natural events has been regulated and started to be used as a metaheuristic algorithm. First, the three strongest wolf groups are determined in the group, and the remaining ones consist of the omega (ω) in the fourth group. There are certain steps in the hunting process. First, the prey is pursued and surrounded by wolves, and they react according to the distance from the prey.

$\overrightarrow{D}$ is the distance between wolf and prey, $\overrightarrow{A}$ is the factor that determines the attack status of the wolf. $\overrightarrow{C}$ is the coefficient factor. For $X_{i,p}$, i shows the design variables, p shows the candidate solution value. Moreover, $\overrightarrow{a}$ is the vector which influences the distance between wolf and prey. According to Eqs. (9–13) that are for $\overrightarrow{D}$, $X_{i,new}$, $\overrightarrow{C}$ as well as $\overrightarrow{A}$ calculate according to the alpha, beta and theta wolves.

$$\vec{D} = \left| \vec{C} \times X_{i,p} - X_{i,j} \right| \tag{9}$$

$$X_{i,new} = X_{i,p} - \vec{A} \times \vec{D} \tag{10}$$

$$\vec{C} = 2 \times rand() \tag{11}$$

$$\vec{A} = 2 \times \vec{a} \times rand() - \vec{a} \tag{12}$$

$$\vec{a} = 2 - 2 \times \frac{t}{maximum\ iteration} \tag{13}$$

After the $X_{i,new}$ is found for all types of wolves, they are averaged (Eq. 14).

$$X_{i,new} = \frac{X_{i,\alpha_{new}} + X_{i,\beta_{new}} + X_{i,\delta_{new}}}{3} \tag{14}$$

For the wolf to attack the prey, the requirement in Eq. (15) must be provided. If this value is not provided, the variables are reassigned based on their maximum minimum value.

$$X_{i,new} = \left|\vec{A}\right| < 1 \tag{15}$$

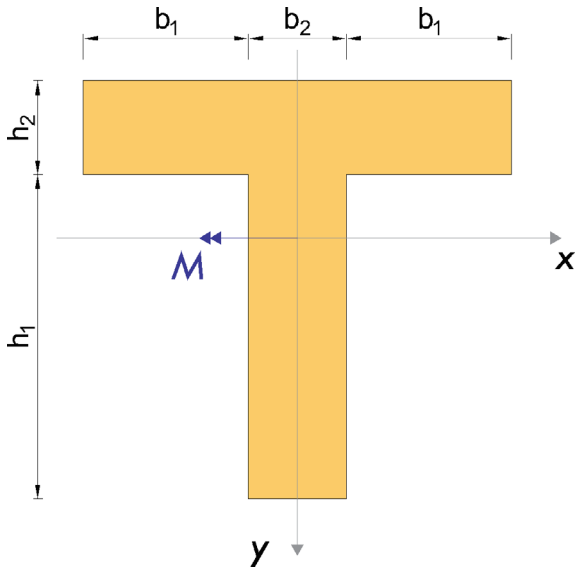
4 Numerical Example

In this study, 2 different shapes were created and the dimensions of the sections were assigned as variable and the dimensions of the sections under certain stresses were found to be cross-sectional area optimization. Each section was found in different sizes and areas because of their different shapes. Table 1 delineates the constants, constraints and variables of the problem for T shape which is demonstrated in Fig. 6 with the cross-sections.

Table 1 The properties of T and I shape systems

Explanation	Unit	Values
$b_{1,min}$	mm	10
$b_{1,max}$	mm	50
$b_{2,min}$	mm	10
$b_{2,max}$	mm	30
$h_{1,min}$	mm	30
$h_{1,max}$	mm	60
$h_{2,min}$	mm	10
$h_{2,max}$	mm	30
Maximum allowable stress	N/mm^2	200 250 300
Moment value	kNm	7.5

Fig. 6 T shape



In order to reach the optimization results, all given properties for the shapes are used in the algorithm which provides to solve in terms of objective function types for the problems.

For Table 2 which shows the results of T shape, the distribution of moment in the system is distributed according to centre of gravity as stress. Therefore, it can clearly be seen that the maximum and minimum stress in the system is calculated different amount because of the shape of system. It is seen that less stresses occur in the region close to the centre of gravity, and more stresses occur in the region far from the centre of gravity (Figs. 7 and 9). Shape differences also affect the minimum value of the area to be optimized, as well as the differences in section lengths for each part in the figure.

Figure 8 demonstrates the I shape which is exploited for the design of structural elements according to the effect of moment.

Looking at Table 3 in more detail, when optimized values for I and T sections are checked, it is observed that under the effect of different allowable stresses, the I section will have a lower area between 138 and 93% compared to the T section. It is known that this difference is due to the shape of the cross-section. Since operations such as moment of inertia and centre of gravity are directly related to the shape of the cross-section, there will be differences in minimum area values between the two shapes.

Table 2 The problem results in terms of maximum allowable stress for T shape

Explanation	Unit	Values according to maximum allowable stress		
		200	250	300
b_1	mm	10	10	10
b_2	mm	24.09	18.8	16.57
h_1	mm	60	60	60
h_2	mm	30	30	26.87
Objective function	mm^2	2768	2292	1977
Maximum stress	N/mm^2	200	250	300
Minimum stress	N/mm^2	−149.5	−175.7	−205.2
Moment of inertia (I_x)	mm^4	1.93×10^6	1.59×10^6	1.29×10^6
Moment of inertia (I_y)	mm^4	1.68×10^6	1.34×10^6	1.06×10^6
Product moment of inertia (I_{xy})	mm^4	0	0	0
Bottom section modulus (W_{bottom})	mm^3	37,500	30,000	25,000
Top section modulus (W_{top})	mm^3	−50,169	−42,686	−36,559

Fig. 7 T shape stresses

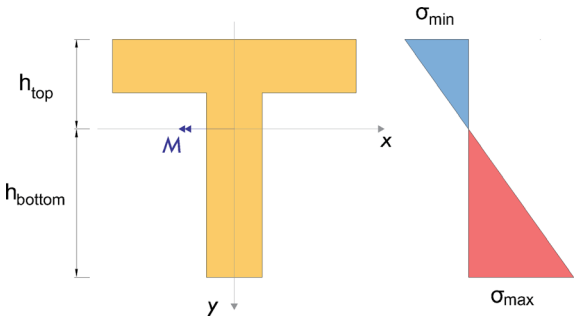


Fig. 8 I shape

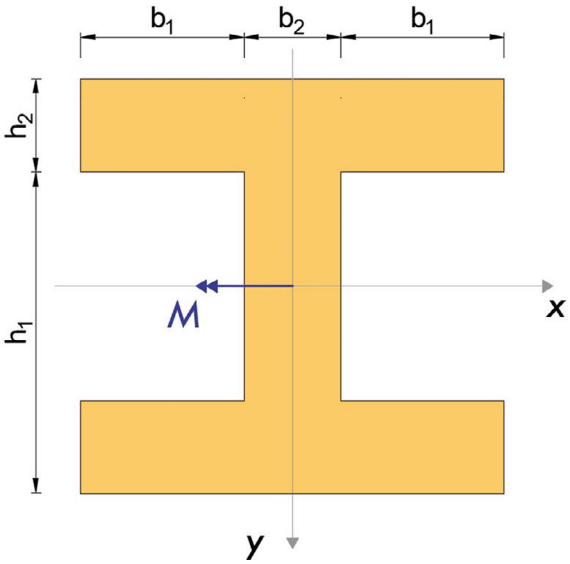


Fig. 9 I shape stresses

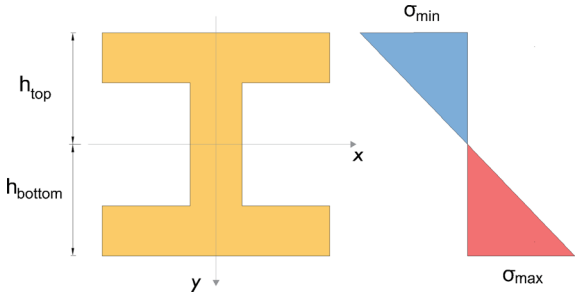


Table 3 The problem results in terms of maximum allowable stress for I shape

Explanation	Unit	Values according to maximum allowable stress		
		200	250	300
b_1	mm	10	10	10
b_2	mm	10	10	10
h_1	mm	56.27	48.47	42.73
h_2	mm	10	10	10
Objective function	mm^2	1163	1085	1027
Maximum stress	N/mm^2	200	250	300
Minimum stress	N/mm^2	−200	−250	−300
Moment of inertia (I_x)	mm^4	1.43×10^6	1.03×10^6	0.78×10^6
Moment of inertia (I_y)	mm^4	1.33×10^6	$0.98-10^6$	0.76×10^6
Product moment of inertia ($I_{xy)}$	mm^4	0	0	0
Bottom section modulus (W_{bottom})	mm^3	37,500	30,000	25,000
Top section modulus (W_{top})	mm^3	−37,500	−30,000	−25,000

5 Conclusion

The usage of bending elements can be various, and they are used in the civil engineering area to create sufficient strength, which provides to the system to remain in control. This strength can change according to the elements which are employed. Therefore, whenever engineers design the structure in terms of the effect of the loadings such as live, dead or seismic load, they need to choose the appropriate design and materials to remain safe side for construction and prevent immediate collapse. Bending control in the system has a significant position throughout the design process because these elements provide flexibility to the system in terms of the direction of moment or loading. The direction of the moment affects the compression tension area, and it is observed the decreasing fibres in the compression zone. However, it is explicitly seen in the increasing fibres gap in the tension zone. Therefore, these situations influence the direction and values of the stress. In the example, there is three minimum allowable stress for the shapes (I and T shape) to reach the minimum area of shapes. Grey wolf algorithm which is the one of metaheuristic algorithms was employed for the optimization of the area. After the optimization results were obtained, values for the shapes were delineated in the tables to control differences not only according to allowable stress but also in terms of shapes. In order to reach the minimum area, algorithms choose the maximum value of the stress in the system for all shapes in at least one region. The variables were specified randomly and regarding GWO equations. After that, the calculation of the system was completed and the final results were found. It can be seen that the minimum area was found for the I shape compared T shape due to the different shapes. Different centres of gravity and the moment of inertia impact the results directly. As for 200 maximum allowable stress,

T and I shape objective function values were found as 2768 and 1163 mm^2, respectively. Regarding 250 maximum allowable stress, T and I shape objective function values were found as 2292 and 1085 mm^2, respectively. Regarding 300 maximum allowable stress, T and I shape objective function values were found as 1977 and 1027 mm^2, respectively. As a consequence, differences between these two shapes can change concerning the allowable stress from 138 to 93%.

References

1. Hill, R.: The Mathematical Theory of Plasticity, pp. 287, 292. Oxford University Press, London (1950)
2. Timoshenko, S.P., Goodier, J.N.: Theory of Elasticity. Mc Graw-Hill, New York (1970)
3. Alexander, J.M.: An analysis of the plastic bending of wide plate and the effect of stretching on transverse residual stresses. Proc. Instn. Mech. Engrs. **173**(1) (1959)
4. Elchalakani, M., Xiao-Ling, Z., Grzebieta, R.H.: Concrete-filled circular steel tubes subjected to pure bending. J. Constr. Steel Res. **57**(11), 1141–1168 (2001)
5. Zibasokhan, H., Behnamfar, F., Azhari, M.: Experimental study of a new pure bending yielding dissipater. Bull. Earthq. Eng. **17**, 4389–4410 (2019)
6. Pham, C.H., Hancock, G.J.: Experimental investigation and direct strength design of high-strength, complex C-sections in pure bending. J. Struct. Eng. **139**(11), 1842–1852 (2013)
7. Magnucki, K., Paczos, P.: Theoretical shape optimization of cold-formed thin-walled channel beams with drop flanges in pure bending. J. Constr. Steel Res. **65**(8–9), 1731–1737 (2009)
8. Naik, N.K., Kumar, R.R., Rajaiah, K.: Optimum hole shapes in beams under pure bending. J. Eng. Mech. **112**(4), 407–411 (1986)
9. Qiao, H., Li, H.: The discussion on optimization models of pure bending beam. Int. J. Adv. Struct. Eng. **5**, 1–10 (2013)
10. Philpot, T.A.: Mechanics of Materials: An Integrated Learning System, 4th edn. Wiley (2017)
11. Subramanian, R.: Strength of Materials, vol. 2. Oxford University Press (2010)
12. Beer, F.P., Johnston, E.R., DeWolf, J.T., Mazurek, D.F., Sanghi, S.: Mechanics of Materials, vol. 1. MC Graw-Hill, New York (1992)
13. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2010)
14. Agrawal, P., Abutarboush, H.F., Ganesh, T., Mohamed, A.W.: Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019). IEEE Access **9**, 26766–26791 (2021)
15. Gandomi, A.H., Yang, X.S., Talatahari, S., Alavi, A.H.: Metaheuristic algorithms in modeling and optimization. In: Metaheuristic Applications in Structures and Infrastructures, vol. 1 (2013)
16. Blum, C., Roli, A.: Metaheuristic in combinatorial optimization: overview and conceptual comparison. ACM Comput. Surv. **35**, 268–308 (2003)
17. Gandomi, A.H., Alavi, A.H.: Krill herd: a new bio-inspired optimization algorithm. Commun. Nonlinear Sci. Numer. Simul. **17**(12), 4831–4845 (2012)
18. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley and Sons, Hoboken, NJ (2010)
19. Yang, X.S.: Metaheuristic optimization: algorithm analysis and open problems. In: Proceedings of the 10th Symposium of Experimental Algorithms, 6630, pp. 21–32 (2011)
20. Çoşut, M., Bekdaş, G., Nigdeli, S.M.: Comparison of different regulations and metaheuristic algorithms in beam design. Int. J. Eng. Appl. Sci. **15**(1), 1–18 (2023)
21. Xu, A., Li, S., Fu, J., Misra, A., Zhao, R.A.: Hybrid method for optimization of frame structures with good constructability. Eng. Struct. **276**, 115336 (2023)

22. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
23. Bekdaş G., Nigdeli M.N., Yücel M., Kayabekir A.E.: Yapay Zeka optimizasyon Algoritmaları ve Mühendislik Uygulamaları, Seçkin Yayıncılık, Ankara (2021)

A Simplified Flower Pollination Algorithm for Structural Optimization of Trusses

Gebrail Bekdaş and Sinan Melih Nigdeli

Abstract The flower Pollination Algorithm (FPA) is inspired by the reproduction process of flowering plants and it uses the pollination processes as two phases that represent global and local optimization. In this study, these two phases are combined in a single phase in the process of modification of the flower pollination algorithm. This simplified FPA is applied to a structural optimization problem which is weight minimization of trusses. The results showed that the simplified FPA is also competitive with classical FPA with two phases.

Keywords Flower pollination algorithm · Optimization · Structural optimization · Single-phase · Truss structures

1 Introduction

In the field of civil engineering, the design of structural components often relies on methods inspired by nature and metaheuristic algorithms. The primary objective in such designs is to minimize material costs, which are directly tied to the total weight of the structure. Achieving an economically good design while adhering to various constraints can make the engineering problem highly non-linear. In such cases, the only feasible approach is to employ numerical iterations. To obtain precise solutions with efficient computational costs, there is a significant focus on utilizing swarm intelligence-based methods in ongoing research.

One of the most prominent optimization problems is the sizing of three-dimensional truss structures. Adeli and Kamal, for instance, utilized a dual simplex algorithm for this purpose [1]. Genetic algorithms have also been applied in the optimal design of truss structures [2, 3]. Camp and Bichon explored truss structures

G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_19

using ant colony optimization [4]. Another example of a metaheuristic method is the Big Bang-Big Crunch (BB-BC) algorithm, employed for optimizing truss structures [5]. Li et al. developed a hybrid method that combines a particle swarm optimizer with passive congregation and harmony search for truss structures [6]. The methodology developed by Perez and Behdinan utilizes particle swarm optimization [7]. In the sizing and layout optimization of truss structures, simulated annealing is utilized by Lamberti [8]. Kaveh and Talatahari hybridized BB-BC with particle swarm optimization and sub-optimization mechanisms [9]. Sonmez combined artificial bee colony with an adaptive penalty function for optimizing truss structures [10]. Degertekin used two variations of harmony search (efficient and self-adaptive) for optimizing truss systems [11]. Teaching Learning-Based Optimization (TLBO) was employed by Camp and Farshchin for the optimum design of truss structures [12]. Kaveh et al. used hybrid particle swarm optimization for truss structures [13]. They also employed chaotic swarming of particles in the size optimization of truss systems [14]. Colliding Bodies Optimization (CBO) was applied for truss system optimization [15], and an enhanced version of CBO was used by Kaveh and Ilchi Ghazaan for the same types of problems [16]. The Flower Pollination Algorithm (FPA) developed by Yang [17] was also used by Bekdaş et al. [18] for the sizing optimization of truss structures. Metaheuristic algorithms were modified by Levy Flight's modification by Bekdaş et al. and methods were compared for optimization of truss structure [19]. FPA and Jaya algorithms were hybridized by Yücel et al. to solve truss structure optimization problems [20, 21]. In addition to the optimization, hybrid methods including machine learning in the prediction of optimum results were proposed by Yücel et al. using multilayer perceptions [22].

Metaheuristic methods are modified to adapt to specific problem requirements, improve their performance, handle constraints, incorporate prior knowledge, enable parallelization, create hybrid algorithms, address dynamic problem environments, enhance scalability, align with user preferences, and support benchmarking and research efforts. These modifications help make metaheuristic algorithms more effective and efficient in solving a wide range of optimization problems. In this study, a simplified version of FPA is proposed for structural optimization. For the basic application of the algorithm, a single-phase version combining global and local pollination processes is proposed and the modified algorithm is applied to the truss optimization problem for verification by comparing it with classical FPA.

2 The Flower Pollination Algorithm and Modification

FPA draws its inspiration from the natural process of flower pollination in flowering plants. In the work of Yang [17], the FPA algorithm was developed based on four fundamental principles related to biotic (cross) pollination, abiotic (self) pollination, flower constancy, and the concept of a switch probability. The methodology for applying FPA to optimize truss sizing problems can be elucidated in three key steps.

The initial step, like any other structural optimization problem, involves defining the design constants, establishing the permissible ranges for design variables, and formulating the design constraints. Additionally, specific algorithm constraints associated with FPA, such as the number of flowers (n), the switch probability (sp), and the maximum generation number (gmax), are specified.

Moving on to the second step, the process begins with the random assignment of initial values to the design variables before commencing the iterative optimization procedure. For each set of design variables, structural analyses are carried out to ensure that they comply with the defined design constraints.

Being a metaheuristic algorithm, FPA employs two distinct methods for updating design variables, and the choice between these methods is governed by a switch probability, which dictates the type of modification to be applied. During the global search phase, FPA simulates biotic pollination. In this process, pollinators imitate the behavior of carrying pollen and adhere to Lévy flight rules. Consequently, new design variables $\left(X_i^{t+1}\right)$ are computed based on the previous solution $\left(X_i^t\right)$, the Lévy distribution (L), and the best design variables associated with the maximum weight (g*), as described in Eq. (1).

$$x_i^{t+1} = x_i^t + L\left(x_i^t - g^*\right). \tag{1}$$

During the local optimization phase, FPA employs abiotic pollination, which involves interactions between pollinators within the same plant. In this context, the pollinators correspond to different flowers of the same plant. As outlined in Eq. (2), the generation of $\left(X_i^{t+1}\right)$ is determined by the variable $\in$ (a random number ranging from 0 to 1) and the variables associated with the jth $\left(X_i^t\right)$ and kth $\left(X_k^t\right)$ flowers, which are selected at random.

$$x_i^{t+1} = x_i^t + \in \left(x_j^t - x_k^t\right) \tag{2}$$

The two types of optimization phases are chosen according to sp. In this study, these two phases are joined as a single phase as given in Eq. (3).

$$x_i^{t+1} = x_i^t + (sp)L\left(x_i^t - g^*\right) + (1 - sp) \in \left(x_j^t - x_k^t\right) \tag{3}$$

The iterative process is executed for the maximum predetermined number of generations, and the primary objective of this process is to achieve weight minimization as the optimization objective.

3 Numerical Examples

The structure model consisting of 25 bars is depicted in Fig. 1, and the various loading cases are detailed in Table 1 [18]. For this particular example, the elasticity modulus and density parameters have been set at 10 Msi and 0.1 lb/in^3, respectively. The design variables, which represent the range of cross-sectional areas, are constrained within the interval of 0.01 to 3.4 in^2. It's worth noting that the compression and tension constraints differ for each member, and these specific limits are provided in Table 2. The members have been organized into groups based on these constraint values.

The optimum results for modified FPA (MFPA) and classical FPA are given in Tables 3 and 4, respectively. The verification is done with multiple sp values and randomly defined sp values for all iterations. The tables include minimum (min), mean, standard deviation (std), needed iteration number to reach best optimum (iter num for min) and mean of needed iteration number for 10 independent runs of the optimization process done for a maximum of 1000 iterations.

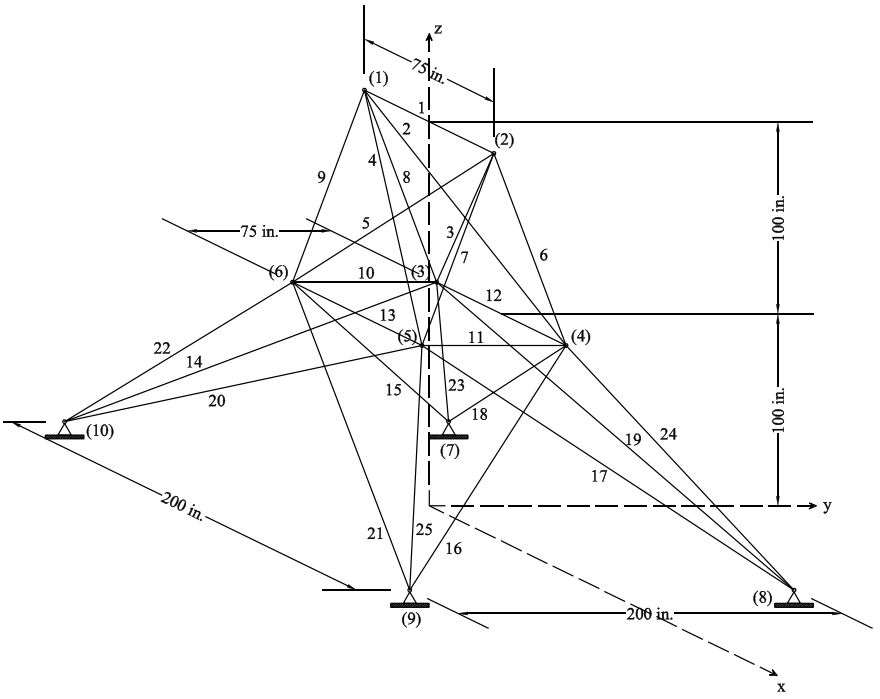


Fig. 1 25-bar truss structure

Table 1 The loading cases of 25-bar structure

Case	Node	P_x (kips)	P_y (kips)	P_z (kips)
1	1	1.0	10.0	−5.0
	2	0.0	10.0	−5.0
	3	0.5	0.0	0.0
	6	0.0	0.0	0.0
2	1	0.0	20.0	−5.0
	2	0.0	−20.0	−5.0

Table 2 The design constraint limits of the 25-bar structure

Members	Compression (ksi)	Tension (ksi)
1	35.092	35
2–5	11.590	35
6–9	17.305	35
10,11	35.092	35
12,13	35.092	35
14–17	6.759	35
18–21	6.959	35
22–25	11.082	35

Table 3 Optimum results of MFPA

sp	min	mean	std	iter num for min	mean iter
0.1	550.880	560.951	8.175	962	763
0.2	553.472	572.236	10.523	844	728
0.3	552.449	573.468	23.152	937	537
0.4	553.193	571.906	11.072	807	697
0.5	552.113	568.195	10.954	650	625
0.6	554.688	568.918	11.414	886	427
0.7	551.032	560.661	6.967	821	583
0.8	546.647	561.768	15.769	408	589
0.9	547.020	562.604	19.723	719	581
1	562.728	580.032	19.503	771	659
rand	547.449	555.161	8.134	660	743

Table 4 Optimum results of FPA

sp	min	mean	std	iter num for min	mean iter
0.1	549.057	554.644	3.472	340	564
0.2	549.615	554.802	2.846	304	474
0.3	549.535	555.061	3.027	753	618
0.4	551.776	554.901	2.439	454	592
0.5	550.057	554.649	4.160	728	669
0.6	552.295	555.097	2.161	960	594
0.7	548.938	557.808	4.996	623	613
0.8	549.953	557.619	4.616	879	753
0.9	548.990	556.711	4.520	721	743
1	555.932	588.338	20.317	447	606
rand	548.400	554.380	3.580	804	673

4 Conclusion

The MFPA demonstrates relatively stable performance across a range of 'sp' values. The 'mean' and 'std' columns show that the mean objective function values and their variations are consistent, indicating that MFPA is robust in handling different parameter settings. MFPA achieves competitive minimum objective function values across various 'sp' settings. This suggests that MFPA is effective at finding good solutions regardless of the specific control parameter value. Although MFPA may not always converge as quickly as FPA, as shown in Table 3, it still requires a reasonable number of iterations to reach the minimum value. This balance between convergence speed and solution quality is an advantage, particularly in problems where computational resources are limited. The 'rand' entry in Table 3 demonstrates that MFPA maintains good performance even when random factors are introduced. This robustness to randomness is advantageous for real-world applications. Also, MFPA can be used as a parameter-free algorithm in that case by using a random number instead of sp in Eq. 3.

Overall, MFPA demonstrates competitive performance across a range of control parameter settings, stability in the presence of randomness, and the ability to handle complex optimization problems. Its adaptability and simplicity make it an attractive choice for various real-world applications. However, the choice between MFPA and other optimization algorithms should depend on the specific characteristics of the problem and the available computational resources.

References

1. Adeli, H., Kamal, O.: Efficient optimization of space trusses. Comput. Struct. **24**(3), 501–511 (1986)
2. Cao, G.: Optimized design of framed structures using a genetic algorithm. Ph.D. thesis, The University of Memphis (1996)
3. Erbatur, F., Hasançebi, O., Tütüncü, I., Kiliç, H.: Optimal design of planar and space structures with genetic algorithms. Comput. Struct. **75**, 209–224 (2000)
4. Camp, C.V., Bichon, B.J.: Design of space trusses using ant colony optimization. J. Struct. Eng. **130**(5), 741–751 (2004)
5. Camp, C.V.: Design of space trusses using big bang–big crunch optimization. J. Struct. Eng. **133**(7), 999–1008 (2007)
6. Li, L.J., Huang, Z.B., Liu, F., Wu, Q.H.: A heuristic particle swarm optimizer for optimization of pin connected structures. Comput. Struct. **85**(7–8), 340–349 (2007)
7. Perez, R.E., Behdinan, K.: Particle swarm approach for structural design optimization. Comput. Struct. **85**, 1579–1588 (2007)
8. Lamberti, L.: An efficient simulated annealing algorithm for design optimization of truss structures. Comput. Struct. **86**, 1936–1953 (2008)
9. Kaveh, A., Talatahari, S.: Size optimization of space trusses using big bang–big crunch algorithm. Comput. Struct. **87**(17–18), 1129–1140 (2009)
10. Sonmez, M.: Artificial bee colony algorithm for optimization of truss structures. Appl. Soft Comput. **11**(2), 2406–2418 (2011)
11. Degertekin, S.O.: Improved harmony search algorithms for sizing optimization of truss structures. Comput. Struct. **92–93**, 229–241 (2012)
12. Camp, C.V., Farshchin, M.: Design of space trusses using modified teaching–learning based optimization. Eng. Struct. **62–63**, 87–97 (2014)
13. Kaveh, A., Bakhshpoori, T., Afshari, E.: An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput. Struct. **143**, 40–59 (2014)
14. Kaveh, A., Sheikholeslami, R., Talatahari, S., Keshvari-Ilkhichi, M.: Chaotic swarming of particles: a new method for size optimization of truss structures. Adv. Eng. Softw. **67**, 136–147 (2014)
15. Kaveh, A., Mahdavi, V.R.: Colliding bodies optimization method for optimum design of truss structures with continuous variables. Adv. Eng. Softw. **70**, 1–12 (2014)
16. Kaveh, A., Ilchi Ghazaan, M.: Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv. Eng. Softw. **77**, 66–75 (2014)
17. Yang, X.S.: Flower pollination algorithm for global optimization. In: Unconventional Computation and Natural Computation, pp. 240–249 (2012)
18. Bekdaş, G., Nigdeli, S.M., Yang, X.S.: Sizing optimization of truss structures using flower pollination algorithm. Appl. Soft Comput. **37**, 322–331 (2015)
19. Bekdaş, G., Yucel, M., Nigdeli, S.M.: Evaluation of metaheuristic-based methods for optimization of truss structures via various algorithms and Lèvy flight modification. Buildings **11**(2), 49 (2021)
20. Yücel, M., Bekdaş, G., Nigdeli, S.M.: Development of a hybrid algorithm for optimum design of a large-scale truss structure. In: Hybrid Metaheuristics in Structural Engineering: Including Machine Learning Applications, pp. 73–86. Cham: Springer Nature Switzerland (2023)
21. Yücel, M., Nigdeli, S.M., Bekdaş, G.: Optimization of truss structures by using a hybrid population-based metaheuristic algorithm. Arab. J. Sci. Eng., 1–16 (2023)
22. Yucel, M., Bekdaş, G., Nigdeli, S.M.: Decision-making model based multilayer perceptrons for estimation of optimum design properties for truss structure. Challenge **7**(4), 170–179 (2021)

Investigation of the Effect of Initial Parameters on the Performance of Metaheuristic Algorithms on a Structural Engineering Problem

Gebrail Bekdaş and Sinan Melih Nigdeli

Abstract Metaheuristic algorithms are the algorithm that formulizes a happening and this formulation depends on an updated solution. To apply these algorithms, an initial solution is needed and it is randomly chosen. This random value has different objective functions that may be close or far from the optimum. In this study, the effect of randomly generated initial values is examined by using several runs. For this investigation, a structural engineering problem that is the cost optimization of reinforced concrete column was investigated by using an adaptive harmony search algorithm.

Keywords Optimization · Reinforced concrete · Metaheuristics · Harmony search · Initial parameters

1 Introduction

Metaheuristics are algorithms utilized to tackle complex problems through a systematic approach that follows a sequence of tasks. These heuristics initially draw upon human thinking processes [1], with subsequent algorithm development drawing inspiration from various sources, such as metaphors. Among the earliest examples is the tabu search algorithm, which relied on human cognition [2]. As the field progressed, it incorporated evaluations [3, 4], numerous processes [5–9], swarm intelligence [10–12], and natural phenomena [13–16] as sources of inspiration. These algorithms prove invaluable when dealing with problems that cannot be straightforwardly solved and often necessitate iterative numerical approaches.

G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_20

In the realm of civil engineering, particularly within structural engineering, numerous challenges align with this problem category, making metaheuristics a prevalent tool for optimization and analysis [17]. A substantial portion of research in this domain centers on optimization problems, encompassing truss structures [18, 19], tuning of structural control systems [20–24], and the optimal design of reinforced concrete (RC) components [25–30]. However, metaheuristics also find applicability in the structural analysis of intricate and non-linear systems [31–35].

Metaheuristics use a random search process in iteration and this situation may affect the performance and convergence of the method. The assignment of initial values also uses randomization. In the present study, the effect of initial values is investigated, because a randomly assigned set of design variables may have an objective function value that may be far or close to the best objective function that is found at the end of the iterative process. To investigate that a structural engineering problem is chosen. The problem is the cost optimization of reinforced concrete (RC) columns. It is optimized via an adaptive harmony search algorithm that is successfully applied to the problem.

2 Methodology

Within this section, we present an optimization procedure employing an Adaptive Harmony Search (AHS) metaheuristic algorithm with the objective of minimizing costs associated with RC columns. The loading and reinforcement specifications for the RC column, subjected to axial (N_z) and bending moment (M_y) forces, are illustrated in Fig. 1.

The Harmony Search (HS) algorithm, which draws inspiration from a musician's quest for the most harmonious combination of notes, was developed by Geem et al. [7]. The optimization process using HS can be distilled into five key steps.

Step 1: In this initial step, design constants, the allowable range for design variables, and algorithm-specific parameters, including the harmony memory size (HMS, also known as population number, pn), initial harmony memory considering rate ($HMCR_{in}$), initial pitch adjusting rate (PAR_{in}), and the stopping criterion for optimization (maximum number of iterations, MI), are defined. In this particular problem, three design variables such as breadth (b), height (h) and reinforcement ratio of column (ρ) are optimized. The ranges are defined in this study in Eqs. (1)–(3).

$$250 \text{ mm} < b < 500 \text{ mm} \tag{1}$$

$$300 \text{ mm} < h < 1000 \text{ mm} \tag{2}$$

$$0.01 < \rho < 0.06 \tag{3}$$

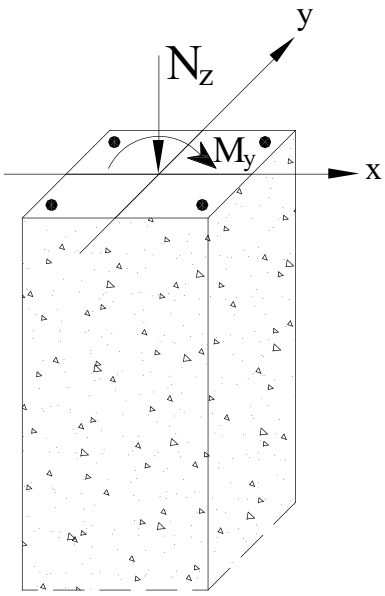


Fig. 1 RC column under uniaxial bending moment

The design constants are as follows: Length of column (l) as 3 m, steel yield strength (f_y) as 420 Mpa, gravity of steel (γ_s) as 7.86 t/m^3, unit cost steel C_c 700 \$/t, concrete compressive streght (f'_c) 30 Mpa, unit concrete cost as 70 \$/m^3, maximum aggregate diameter in concrete (D_{max}) and strain for ultimate concrete stregnt (ε_c) as 16 mm and 0.003. Two cases of loading were considered in the investigation as follows: Flexural moment (M_u) as 100 kNm and 500 kNm, Axial force (N_u) as 1000 kNm and 2000 kNm for Cases 1 and 2, respectively.

Step 2: An initial harmony memory matrix, denoted as HM, is generated according to Eq. (4). This matrix comprises pn candidate solution sets, each encompassing values for each design variable (X_i, where i ranges from 1 to N).

$$\mathrm{HM} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,pn} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,pn} \\ \vdots & \vdots & \ldots & \vdots \\ X_{N-1,1} & X_{N-1,2} & \cdots & X_{N-1,pn} \\ X_{N,1} & X_{N,2} & \cdots & X_{N,pn} \end{bmatrix} \tag{4}$$

In the Harmony Search algorithm, the initial values of the design variables are generated randomly within the defined limits of each variable, which were established in Step 1 as the maximum ($X_{i(max)}$) and minimum ($X_{i(min)}$) allowable values, as expressed in Eq. (5). This random initialization is crucial to ensure a diverse set of initial candidate solutions for the optimization process.

$$X_i = X_{i(\min)} + rand\left(X_{i(\max)} - X_{i(\min)}\right) \tag{5}$$

In Eq. (5), 'rand' represents a random number generated within the range of 0 to 1.

Step 3: In this step, the objective function for each solution set is computed, and the design constraints in accordance with ACI 318: Building Code Requirements for reinforced concrete [36] are verified. For solution sets that fail to meet the specified design constraints, a penalty value is assigned to their objective function values. Subsequently, all the objective function values are stored in a vector. The objective function (OF), defined by Eq. (6), quantifies the total material cost. The aim during the optimization process is to minimize this objective function, which involves searching for an appropriate solution set that achieves this minimum.

$$OF = C_c V_c + C_s W_s \tag{6}$$

In Eq. (6), C_c represents the cost per unit volume of concrete, while C_s stands for the cost per unit weight of reinforcing steel. The volume of concrete is denoted as V_c, and the weight of the reinforcing steel is represented as W_s. The design constraints are given as Eqs. (7)–(11) for Maximum axial force (N_{max}), Minimum steel area (A_{smin}), Maximum steel area (A_{smax}), Flexural strength capacity (M_d) and Axial force capacity (N_d).

$$N_d \leq N_{max} = 0.5 f_c' bh \tag{7}$$

$$A_s \geq A_{smin} = 0.01bh \tag{8}$$

$$A_s \leq A_{smax} = 0.06bh \tag{9}$$

$$M_d \geq M_u \tag{10}$$

$$N_d \geq N_u \tag{11}$$

Step 4: In this step, a new solution set is generated. As per the rules of the Harmony Search (HS) algorithm, the new values for each design variable ($X_{i(new)}$) can be generated using two equations provided in Eqs. (12) and (13). The first equation (Eq. (4)) is akin to the process of generating initial solutions.

$$X_{i(new)} = X_{i(\min)} + rand\left(X_{i(\max)} - X_{i(\min)}\right) \tag{12}$$

The second equation (Eq. (13)) involves the random generation of new values within a range determined by multiplying the Pitch Adjusting Rate (PAR) by the differences between the ultimate limits of the design variable ($X_{i(max)} - X_{i(min)}$). This generation process is applied to a randomly selected solution set.

$$X_{i(new)} = X_{i,k} + rand\ \mathrm{PAR}\left(X_{i(\max)} - X_{i(\min)}\right) \tag{13}$$

In Eq. (13), $X_{i,k}$ represents the value of a design variable within the selected solution set. The choice between the two equations presented in Eqs. (12) and (13) is determined by the Harmony Memory Considering Rate (HMCR). In the Adaptive Harmony Search (AHS) algorithm, both the HMCR and PAR values are updated based on the current iteration (IN) according to the equations given in Eqs. (14) and (15). These updates are performed as the algorithm progresses to adapt to the problem and search space dynamically.

$$PAR = PAR_{in}\left(1 - \frac{IN}{MI}\right) \tag{14}$$

$$HMCR = HMCR_{in}\left(1 - \frac{IN}{MI}\right) \tag{15}$$

Step 5: In this step, comparisons are made between the newly generated solution set and the solutions already present in the existing solution matrix. Specifically, the comparison is based on the objective function values. If the new solution is superior (i.e., it has a lower objective function value) than the worst solution in the existing solution matrix, then the worst solution is replaced by the new solution. However, if the new solution does not outperform the worst solution in terms of the objective function, no modifications are made to the solution matrix.

These last two steps, involving the generation and evaluation of new solutions and their replacement or retention based on objective function comparisons, continue iteratively until the stopping criterion, as defined in the earlier step (Step 1), is satisfied. This iterative process helps the algorithm search for an optimal or near-optimal solution over multiple iterations.

3 Numerical Results

The investigation was done for two cases. The optimum results for multiple runs are the same and given in Tables 1 and 2 for cases 1 and 2, respectively. The investigation includes best initial values, optimum results and parameters, the ratio of initial solution to optimum results and the initial number of that the final optimum result is found.

Also, the convergence plots for iterations are given as Figs. 2 and 3 for cases 1 and 2, respectively.

Table 1 The evaluation of initial parameters for case 1

Initial value	Optimum value	b/h	ρ	Initial/Optimum	Iteration number
84.48	36.33	250/387.5	0.01	2.3	19,444
76.02				2.1	14,636
70.82				1.9	13,368
63.96				1.8	13,561
62.12				1.7	11,834
56.40				1.6	7483
55.61				1.5	4993
51.95				1.4	4754
48.85				1.3	2059
41.96				1.2	1246

Table 2 The evaluation of initial parameters for case 2

Initial value	Optimum value	b/h	ρ	Initial/Optimum	Iteration number
127.6	77.0	250/821.5	0.01	1.7	19,764
108.8				1.4	15,845
106.9				1.4	13,830
97.3				1.3	11,892
93.3				1.2	10,230
82.0				1.1	9511

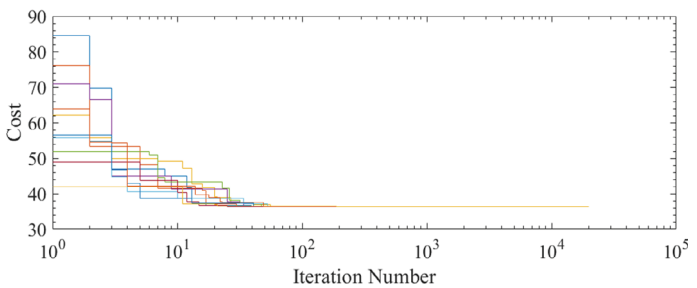


Fig. 2 Convergence plots for case 1

4 Conclusion

For all runs of the optimization process, the best optimum results are obtained although the assigned initial values have ratios of 1.1–2.3 to best optimum results. This proves that the algorithm is effective without trapping to local optimum results. Also, the maximum iterations are 19,444 and 19,764 for cases 1 and 2, respectively.

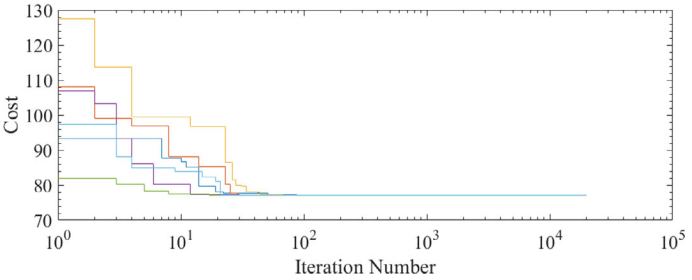


Fig. 3 Convergence plots for case 2

According to convergence plots, higher initial value trials update the first solutions more quickly than the others, but it is clearly seen that if the initial value is close to the optimum, it is quicker to find the optimum solution with minor exceptions.

References

1. Sörensen, K., Sevaux, M., Glover, F.: A History of Metaheuristics. Handbook of Heuristics, pp. 1–18 (2018)
2. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. **13**(5), 533–549 (1986)
3. Goldberg, D.E., Samtani, M.P.: Engineering optimization via genetic algorithm. In: Proceedings of Ninth Conference on Electronic Computation. ASCE, New York, NY, pp. 471–482 (1986)
4. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan (1975)
5. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
6. Erol, O.K., Eksin, I.: A new optimization method: big bang–big crunch. Adv. Eng. Softw. **37**(2), 106–111 (2006)
7. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**, 60–68 (2001)
8. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput.-Aided Des. **43**(3), 303–315 (2011)
9. Rao, R.: Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. **7**(1), 19–34 (2016)
10. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. IV, November 27-December 1, pp. 1942–1948, Perth Australia (1995)
11. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybernet B **26**, 29–41 (1996)
12. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. **39**(3), 459–471 (2007)
13. Yang, X.S., Deb, S.: Engineering optimisation by cuckoo search. Int. J. Math. Model. and Numer. Optim. **1**(4), 330–343 (2010)
14. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimisation. Int. J. Bio-Inspired Comput. **2**(2), 78–84 (2010)

15. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74. Springer, Berlin, Heidelberg (2010)
16. Yang, X.S.: Flower pollination algorithm for global optimization. In: International Conference on Unconventional Computing and Natural Computation, pp. 240–249. Springer, Berlin, Heidelberg (2012)
17. Toklu, Y.C., Bekdas, G., Nigdeli, S.M.: Metaheuristics for Structural Design and Analysis. John Wiley & Sons (2021)
18. Talatahari, S., Goodarzimehr, V.: A discrete hybrid teaching-learning-based optimization algorithm for optimization of space trusses. J. Struct. Eng. Geo-Tech. **9**(1), 0–0 (2019)
19. Bekdaş, G., Yucel, M., Nigdeli, S.M.: Evaluation of metaheuristic-based methods for optimization of truss structures via various algorithms and Lèvy flight modification. Buildings **11**(2), 49 (2021)
20. Bekdaş, G., Nigdeli, S.M., Yang, X.S.: A novel bat algorithm based optimum tuning of mass dampers for improving the seismic safety of structures. Eng. Struct. **159**, 89–98 (2018)
21. Yucel, M., Bekdaş, G., Nigdeli, S.M., Sevgen, S.: Estimation of optimum tuned mass damper parameters via machine learning. J. Build. Eng. **26**, 100847 (2019)
22. Ulusoy, S., Bekdas, G., Nigdeli, S.M.: Active structural control via metaheuristic algorithms considering soil-structure interaction. Struct. Eng. Mech. **75**(2), 175–191 (2020)
23. Ulusoy, S., Nigdeli, S.M., Bekdaş, G.: Novel metaheuristic-based tuning of PID controllers for seismic structures and verification of robustness. J. Build. Eng. **33**, 101647 (2021)
24. Ulusoy, S., Bekdaş, G., Nigdeli, S.M., Kim, S., Geem, Z.W.: Performance of optimum tuned PID controller with different feedback strategies on active-controlled structures. Appl. Sci. **11**(4), 1682 (2021)
25. Gil-Martin, L.M., Hernandez-Montes, E., Aschheim, M.: Optimal reinforcement of RC columns for biaxial bending. Mater. Struct. **43**, 1245–1256 (2010)
26. Camp, C.V., Pezeshk, S., Hansson, H.: Flexural design of reinforced concrete frames using a genetic algorithm. J. Struct. Eng.-ASCE **129**, 105–11 (2003)
27. Govindaraj, V., Ramasamy, J.V.: Optimum detailed design of reinforced concrete frames using genetic algorithms. Eng. Optimiz. **39**(4), 471–494 (2007)
28. Ceranic, B., Fryer, C., Baines, R.W.: An application of simulated annealing to the optimum design of reinforced concrete retaining structures. Comput. Struct. **79**, 1569–1581 (2001)
29. Camp, C.V., Akin, A.: Design of retaining walls using big bang–big crunch optimization. J. Struct. Eng.-ASCE **138**(3), 438–448 (2012)
30. Talatahari, S., Sheikholeslami, R., Shadfaran, M., Pourbaba, M.: Optimum design of gravity retaining walls using charged system search algorithm. Math. Probl. Eng. **2012**, Article ID 301628 (2012)
31. Bekdaş, G., Kayabekir, A.E., Nigdeli, S.M., Toklu, Y.C.: Advanced energy-based analyses of trusses employing hybrid metaheuristics. Struct. Design Tall Spec. Build. **28**(9), e1609 (2019)
32. Toklu, Y.C., Bekdaş, G., Yücel, M., Nigdeli, S.M., Kayabekir, A.E., Kim, S., Geem, Z.W.: Total potential optimization using metaheuristic algorithms for solving nonlinear plane strain systems. Appl. Sci. **11**(7), 3220 (2021)
33. Toklu, Y.C., Kayabekir, A.E., Bekdaş, G., Nigdeli, S.M., Yücel, M.: Analysis of plane-stress systems via total potential optimization method considering nonlinear behavior. J. Struct. Eng. **146**(11), 04020249 (2020)
34. Toklu, Y.C., Bekdaş, G., Kayabekir, A.E., Nigdeli, S.M., Yücel, M.: Total potential optimization using metaheuristics: analysis of cantilever beam via plane-stress members. In: International Conference on Harmony Search Algorithm, pp. 127–138. Springer, Singapore (2020)
35. Kayabekir, A.E., Toklu, Y.C., Bekdaş, G., Nigdeli, S.M., Yücel, M., Geem, Z.W.: A novel hybrid harmony search approach for the analysis of plane stress systems via total potential optimization. Appl. Sci. **10**(7), 2301 (2020)
36. ACI Committee, American Concrete Institute, & International Organization for Standardization. Building code requirements for structural concrete (ACI 318–05) and commentary. American Concrete Institute (2008)

Geospatial Multi-criteria Decision Framework for Municipal Landfill Site Selection in Surat City, India

Shobhit Chaturvedi, Preyansh Patel, Momin Mohammed Misam, Naimish Bhatt, and Dhruvesh Patel

Abstract The fast overhaul of existing urban landfills necessitates scientific approaches to identify new municipal landfill sites. To this end, an integrated Geographic Information System (GIS)—Multi Criteria Decision-Making (MCDM) framework is presented encompassing thirteen (Land Use Land Cover, Built-Up Proximity, Ground Elevation, Ground Slope, Road Proximity, Rail Proximity, Airport Proximity, Land Value, Vegetation Proximity, Water Body Proximity, Land Surface Temperature, Lineament Proximity, Ground Water Depth) Evaluation criteria to propose new municipal landfill sites for Surat city in western Indian state of Gujarat, India. Furthermore, an Analytical Hierarchy Process (AHP) is applied based on assessments from 10 experts to develop a landfill suitability map, integrating the relative contributions of the thirteen evaluation criteria. Out of 1250 sq. km, 10.89% is most suitable, 42.06% is fairly suitable, 26.25% is moderately suitable, 18.23% is less suitable, and 2.57% is unsuitable for landfill site development. Further, TOPIS MCDM technique is applied to rank 15 candidate landfill locations within the most suitable zones. Locations in Vanz, Sachin, Sarol and Kamrez are the top three ranked locations for future landfill development to cater Surat municipal waste. These findings provide valuable insights for urban planners and decision-makers, enabling them to transition from heuristic to evidence-based approaches in landfill site selection.

Keywords Urban landfills · GIS-MCDM framework · Analytic hierarchy process · TOPSIS · Sustainability

S. Chaturvedi (✉) · P. Patel · M. M. Misam · N. Bhatt · D. Patel
Department of Civil Engineering, School of Technology, Pandit Deendayal Energy University Gandhinagar, Gandhinagar, Gujarat, India
e-mail: shobhit.chaturvedi@sot.pdpu.ac.in

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_21

1 Introduction

As urban areas expand, there is a proportional rise in the volume of municipal solid waste (MSW), including various materials from homes, businesses, and institutions [14, 20]. MSW comprises materials discarded as a result of human activities, encompassing paper, plastics, metals, glass, organic waste, and other non-biodegradable substances [28, 29]. Typically, cities resort to landfills as a primary method for managing their MSWs, aiming to safeguard public health and uphold environmental sustainability. However, improper landfill site selection and waste management poses several challenges [3, 16]. This includes environmental contamination, groundwater and soil pollution, and the release of harmful gases like carbon-dioxide, methane and nitrous-oxides contributing to air pollution. Poorly managed landfills also create unsightly and potentially hazardous waste disposal sites, impacting aesthetics and community well-being [2, 25]. Additionally, waste dumps also act as fire hazards during the dry summer season [23, 27]. As populations grow and consumerism increases, the amount of waste being produced is growing at an unprecedented rate, straining landfills and waste management systems. Additionally, the improper handling of waste in landfills depletes valuable land resources posing long-term risks to public health [7, 21]. These issues are even more alarming for developing countries, where waste management infrastructure is often inadequate.

Several factors associated with topography, geology, environment, and infrastructure are critical in the choice of ideal landfill sites [2]. Elements of topography, such as elevation, slope, and land gradient, work to reduce the likelihood of soil erosion and landslides. Geological factors, encompassing land stability, geological formations, and seismic hazards, bolster resilience against natural disasters and geological instabilities that could result in water contamination [15]. Land valuation holds significance, as investing in land in high-value areas may not yield proportional returns for landfill purposes. Site selection is also influenced by environmental factors like the regional distribution of land use, land surface temperatures, and proximity to water bodies. Further, the evaluation of infrastructure services, considering proximity to transportation networks, access to utilities such as water and electricity, and distance from residential zones, is crucial for ensuring efficient waste transportation and operational functionality while minimizing societal disruptions [7].

Assessing these criteria simultaneously is essential when choosing new landfill locations. Remote Sensing (RS) and Geographic Information Systems (GIS) facilitates the retrieval and processing of data for landfill site selection criteria [23, 24]. Different RS data platforms hosted by LANDSAT, SENTINEL, NASA and European Space Agency and ISRO provide free raster datasets linked to several geology, topography and environmental parameters [17]. Besides, online platforms such as DIVAGIS, Natural Earth, Map-box and Open Street Maps provide several key vector layers for locating key infrastructure facilities [27]. GIS tools like ArcGIS and Q-GIS develop thematic raster layers for landfill site selection criteria and can create a unified landfill suitability raster [15]. The estimation of criteria weightages involves the application of suitable Multi Criteria Decision Making Techniques

such as Analytical Hierarchy Process (AHP). AHP offers a systematic framework to incorporate expert views while assessing several complementary and conflicting criteria to develop an overall landfill site suitability raster [18].

The past few years witnessed a considerable rise in the published literature on the adoption of GIS-MCDM frameworks for selecting optimal locations for prospective landfill development [4, 10–14, 30). Özkan et al. [17] Özkan et al. [17] reviewed 106 studies from 2005–2019, classifying landfill criteria and identifying research gaps, emphasizing AHP and WLC in GIS-based MCDM modelling for landfill site suitability analysis. Environmental aspects, especially surface water, are prominently studied. Jamshidi et al. [11] employed employs ANP, fuzzy logic, OWA, and WLC in GIS to select a landfill site for Markazi province Iran, revealing the effectiveness of the integrated approach in providing flexible and improved site selection compared to traditional methods. Researchers also applied GIS-MCDM models for optimal landfill locations in Indian cities such as Thiruverumbur [1, 19], Prayagraj [27] and Kolkata [23]. Sk et al. [25] addressed solid waste disposal challenges in Durgapur, India, utilizing GIS and AHP. Eleven criteria, including land elevation, slope, and land cost, were assessed to identify three potential landfill areas. Land cost emerged as the most significant criterion in the model. Ali et al. [2] utilized GIS and fuzzy MCDM for landfill site selection in Memari Municipality, identifying 7 suitable sites. Roy et al. [21] adopted GIS and AHP for multi-criteria decision analysis, this study identified landfill sites in the Siliguri municipal corporation area. Dolui & Sarkar [7] addressed landfill site selection challenges in Kharagpur, India, utilizing MCDM methods (AHP, Fuzzy AHP, SRS, RSW).

Existing landfill sites in many Indian cities have surpassed their service life, posing risks of environmental degradation, fires, and air pollution. This study proposes an integrated GIS-MCDM framework to identify new landfill sites for Surat, a large city in the western state of Gujarat, India. Using the AHP process, all relevant landfill selection criteria are integrated to create an overall landfill suitability map for the Surat region. Additionally, the TOPSIS method is used to rank candidate landfill sites, pinpointing the most optimal locations. These findings will aid urban planners in focusing on sustainable waste management practices.

2 Study Area

Surat, located in the state of Gujarat in Western India, is a busy city with a rich history and a rapidly growing urban landscape. Situated along the banks of the Tapi River, Surat boasts a diverse geography, including plains, hills, and agricultural lands. The city is positioned between 21°09′ and 21°23′ north latitude and 72°40′ and 73°02′ east longitude, falling within toposheet numbers 47-I and 47-J according to the Survey of India. Figure 1a, b depict the study area, covering 1250 sq. km around Surat city.

Surat has witnessed rapid urbanization and population growth in recent years, leading to increased demand for infrastructure and services. This growth has also resulted in a significant rise in municipal solid waste (MSW) generation. The city

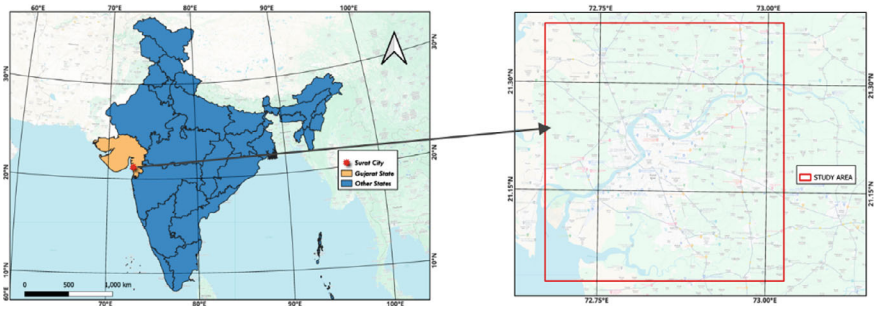


Fig. 1 **a** Geographical Location of Surat city in India Map; **b** Study area selected for analysis

Fig. 2 Environmental challenges associated to pollution and fire at the Khajod landfill in Surat

currently generates approximately 2,100–2,300 tons of MSW daily across its 60 wards. The Khajod landfill was a major waste landfill site in Surat but was closed in February 2020 on order from the National Green Tribunal due to environmental issues such as pollution and fires shown in Fig. 2. As of now, the city does not have adequate landfill space to accommodate the projected increase in waste generation. Therefore, there is a need for systematic evaluations to identify suitable new landfill sites for safe MSW disposal.

3 Methodology

Landfill Evaluation Criteria

As illustrated in Fig. 3, a comprehensive research methodology was devised for selecting suitable landfill sites across various Surat city. The initial phase involved identifying relevant criteria for landfill site selection from existing literature. Thirteen criteria were considered for landfill site selection in this study, including Land Use Land Cover, Built-Up Proximity, Ground Elevation, Ground Slope, Road Proximity, Rail Proximity, Airport Proximity, Land Value, Vegetation Proximity, Water

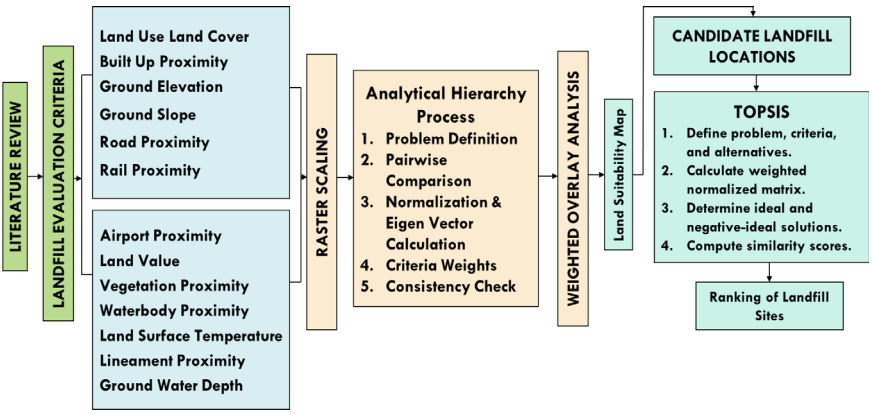


Fig. 3 A Comprehensive research framework for municipal landfill site selection in Surat, Gujarat, India

Body Proximity, Land Surface Temperature, Lineament Proximity and Ground Water Depth.

Table 1 provides the justification for selecting these specific criteria. The raster and vector datasets for these criteria were obtained from various sources, as listed in Table 2. The thematic raster layers for the selected criteria were developed using the Quantum GIS (QGIS) software. The ground elevation data was sourced from NASA's SRTM-Global Digital Elevation Model and utilized to derive the ground slope map for the region. The Near-Infrared (NIR) band of LANDSAT was used for Land Surface Temperature (LST) mapping. The LULC map came from ESRI's Global Land Cover dataset, derived from ESA Sentinel 2 imagery. QGIS proximity raster tools generated Barelands, Built-Up, and Water-Body proximity maps. Road and rail data were sourced from DIVAGIS and rasterized for proximity analysis. Land value rates were collected from online property websites, converted into vector format, and then rasterized using the nearest neighbour interpolation method. Airport campus location was obtained from Google Maps for proximity assessment. Groundwater and lineament data were acquired from the Central Ground Water Board and Geological Survey of India portals, respectively.

Thematic raster layers for the thirteen landfill criteria were classified into five suitability classes for landfill site selection. Seven criteria (RaP, LiP, LULC, BUP, WBP, AP, VP) were identified as favourable, while six (LV, G-El, GS, RoP, LST, GWD) were categorized as unfavourable, indicating a preference for higher values in beneficial attributes and lower values in non-beneficial ones.

Analytic Hierarchy Process

Introduced by Saaty in 1987, [22] Analytic Hierarchy Process (AHP) is a robust Multi Criteria Decision Making method, employing pairwise comparisons for decision-making. Its hierarchical structure and mathematical computations enable rational

Table 1 The selection criteria for landfill sites and the logic behind their selection

Landfill criteria	Justification	References
Land Use Land Cover (LULC)	Land Use Land Cover (LULC) is crucial for landfill site selection, favouring barren lands while urban areas and water bodies are deemed unsuitable due to contamination risks	[10]
Built-Up Proximity (BUP)	Urban landfills cause odour, noise, and health risks. It's crucial to position them away from built-up areas	[21, 24]
Ground Elevation (GEl)	Flat terrains are ideal for landfills, minimizing contamination risks. Steep slopes at higher elevations intensify issues, making such areas less suitable for landfills	[21, 24]
Ground Slope (GS)	Ground slope significantly impacts landfill functionality by influencing runoff, erosion, and waste dispersion, with gentle slopes proving more effective in containing waste and mitigating environmental risks compared to steeper slopes	[10, 25]
Road Proximity (RoP)	Close proximity to roads is vital for efficient waste transport, enhancing landfill suitability by ensuring smoother waste transportation from urban areas	[2, 17]
Rail Proximity (RaP)	Selecting landfill sites distant from railways ensures better suitability by mitigating noise, safety, and potential environmental concerns associated with their proximity	[9, 27]
Airport Proximity (Ap)	Landfill sites should be located far from airports to prevent bird strikes, comply with height restrictions, and minimize environmental impact, including odour and emissions	[24]
Land Value (LV)	Using costly city lands for landfills is impractical due to limited availability and high costs. High-value areas may not yield proportional returns for landfill investment	[6, 21, 27]
Vegetation Proximity (VP)	Urban vegetation, including trees and shrubs, is crucial for urban ecosystems. Landfills should be located far from these areas to prevent contamination and maintain their ecological functions	[24]
Water Body Proximity (WBP)	Indian regulations strictly forbid waste disposal near water bodies due to contamination risks, as landfills near streams or canals can impact surface and sub-surface water	[5, 30]

(continued)

Table 1 (continued)

Landfill criteria	Justification	References
Land Surface Temperature (LST)	Areas with lower LSTs are favoured due to potentially reduced waste decomposition rates, emissions, and associated environmental impacts, making them more suitable choices for landfill placement	[1, 23, 26]
Lineament Proximity (LiP)	Lineaments, linear faults on the Earth's surface, impact rock permeability and porosity. Some of these fractures may intersect with groundwater, posing pollution risks. Landfills should be distant from these areas to mitigate pollution risks	[8, 21]
Ground Water Depth (GWD)	Landfills are typically sited in areas with deep groundwater to prevent pollution of drinking water sources. Regular monitoring is essential to ensure groundwater remains uncontaminated	[5, 30]

Table 2 Thematic layers for landfill site selection, specifications, and data sources

Landfill criteria	Data details	Data source
Land use land cover	Raster, 10 m resolution	ESRI Land Cover
Built-up proximity		
Water-body proximity		
Ground elevation	Raster, 10 m resolution	NASA Earth Data
Vegetation Proximity		
Ground slope		
Land value	Vector, Point Data	Property Websites (99acres, Makaan, Housing etc.)
Lineament	Vector, Polygon Data	Geological Survey of India Website
Land surface temperature	Raster, 30 m resolution	USGS Earth Explorer
Road proximity	Vector, Line Layer	DIVAGIS, Open Street Map
Rail proximity		
Ground water	Vector, Borehole point data	Central Ground Water Board

decision synthesis in complex problem-solving for engineering, technology, and management.

1. **Problem Definition**: It involves defining the decision problem and selection of appropriate goals.
2. **Criteria Pairwise Comparison**: It involves assessing the relative importance of criteria through pairwise comparisons using Saaty's scale as shown in Table 3 (1 to 9) to generate a square matrix denoted as $\overline{A}$.

Table 3 Analytic Hierarchy Process criteria pairwise comparison scale [22]

Less important				Equally important	More important			
Extremely	Very strongly	Strongly	Moderately		Moderately	Strongly	Very strongly	Extremely
1/9	1/7	1/5	1/3	1	3	5	7	9

Ten subject matter experts, highlighted in Table 4 were engaged to obtain the criteria wise pairwise comparison data. The panel of experts involved professionals, researchers and academicians associate with urban solid waste management and sustainability.

3. **Normalization and Eigen Vector Calculation**: The normalized matrix W is computed from $\overline{A}$ by dividing each element by its column sum, ensuring a sum of 1 for each column. The resulting matrix W represent pairwise comparisons. The principal eigenvectors V are derived by averaging the rows in W and the maximum eigenvalue (λ_{max}) are computed to indicate criteria dominance and relative importance, calculated as per Eq. 1. by summing the elements of V and dividing by the matrix order (n).

$$\lambda_{\max} = \frac{\sum V}{n} \tag{1}$$

4. **Consistency Check: T**he reliability of pairwise comparisons in the AHP process is evaluated using the Consistency Index (CI) (refer Eq. 2) and Consistency Ratio (CR) (refer Eq. 3).

Consistency Index (CI):

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{2}$$

The CI value signifies the consistency of judgments in pairwise comparisons.

Consistency Ratio (CR):

$$CR = \frac{CI}{RI} \tag{3}$$

Table 4 Designation and experience details for the contacted experts

Respondent	Designation	Experience
R 1	Asst Prof, Civil Engineering	7 years
R 2	Associate Prof, Urban Planning	12 years
R 3	Town Planning, Surat Municipal Corporation (SMC)	15 years
R 4	Engineer, SMC	8 years
R 5	Urban Planner, SMC	6 years
R 6	Town Planner, SMC	5 years
R 7	Inspector, SMC	7 years
R 8	Site Engineer, SMC	15 years
R 9	Asst. City Planner, SMC	30 years
R 10	Inspector, SMC	12 years

The CR compared CI with a predefined RI value (1.45 for AHP with 10 criteria). As per Table 5, if CR was below 0.10, judgments were deemed consistent. Otherwise, experts reassessed pairwise comparisons for improved consistency, ensuring reliable decision-making.

Weighted Overlay Analysis

In the Weighted Overlay Analysis step, the QGIS Raster Calculator Tool was used to combine the individual criteria raster layers using their weights from Eq. 4, resulting in the computation of the Landfill Suitability Index.

$$\text{Landfill Suitability Index (LSI)} = \sum_{i=1}^{n} w_i . x_i \tag{4}$$

where w_i signifies the weight of the ith criterion and x_i refer to the pixel-wise scaled raster value for each criterion obtained after the raster scaling stage. LSI raster layer represents the varying degrees of landfill site suitability across the study area. Lower LSI values represents inferior suitability, whereas higher scores indicate a higher suitability for landfill site development. The LSI range was split into five bands consistent with 'Very Low', 'Low', 'Moderate', 'High' and 'Very High' suitability for landfill site development. The LSI zoning, though useful, may not completely fulfil landfill development requirements. Hence several landfill sites across Surat city, all in high LSI zones, were identified for further evaluation. The TOPSIS MCDM method was used to assess these sites and determine the most optimal location. TOPSIS aims to select the alternative with the shortest distance to the positive ideal solution and the farthest distance from the negative ideal solution.

TOPSIS Based Landfill Site Ranking

1. **Problem Definition**: It involves defining the problem and identifying key criteria and potential sites for landfill development. Extract evaluation criteria values for all sites to create a decision matrix.
2. **Normalize the Decision Matrix**: This involves creating a decision matrix (X) where rows represent sites and columns represent criteria. Normalize the matrix to standardize all criteria values between 0 and 1 using the formula in Eq. 5.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x^2{}_{ij}}} \tag{5}$$

Table 5 AHP random consistency index [22]

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.32	1.41	1.45	1.49	1.49

Here, r_{ij} is the normalized value for site i and criterion j, and x_{ij} is the original value in the decision matrix.

3. **Calculate Weighted Normalized Decision Matrix**: The weighted decision matrix (W) was normalized as per Eq. 6 by multiplying the normalized decision matrix (R) with the criteria weights (W_j) to obtain the weighted normalized decision matrix (V):

$$\boldsymbol{V_{ij} = r_{ij}.W_j} \tag{6}$$

Here, V_{ij} is the value in the weighted normalized decision matrix for site i and criterion j, r_{ij} is the normalized value, and W_j is the weight of criterion j.

4. **Identify Ideal and Anti-Ideal Solutions**: The ideal solution ($\boldsymbol{A}^+$) and anti-ideal solution ($\boldsymbol{A}^-$) were determined by identifying the best and worst values for each criterion using Eqs. 7 and 8.

$$\boldsymbol{A}^+ = (\max(V_1), \max(V_2), \ldots, \max(V_m)) \tag{7}$$

$$\boldsymbol{A}^- = (\min(V_1), \min(V_2), \ldots, \min(V_m)) \tag{8}$$

Here, V_j represents the column vector of the weighted normalized decision matrix for criterion j.

5. **Calculate Distance to Ideal and Anti-Ideal Solutions**: The Euclidean distance between each site and both the ideal and anti-ideal solutions was calculated using Eqs. 9 and 10.

$$\text{Distance to Ideal Solution } (S_i{}^+) = \sqrt{\sum_{j=1}^{m} (V_{ij} - A_j{}^+)^2} \tag{9}$$

$$\text{Distance to Anti} - \text{Ideal Solution}(S_i{}^-) = \sqrt{\sum_{j=1}^{m} (V_{ij} - A_j{}^-)^2} \tag{10}$$

6. **Determine Proximity to the Ideal Solution**: Calculate the proximity measure (closeness coefficient) for each site using the Eq. 11.

$$\boldsymbol{C_i} = \frac{\boldsymbol{S_i}^-}{\boldsymbol{S_i}^+ + \boldsymbol{S_i}^-} \tag{11}$$

7. **Rank and Select Sites**: Rank the sites based on their closeness coefficients. The site with the highest closeness coefficient was considered the most optimal for landfill development.

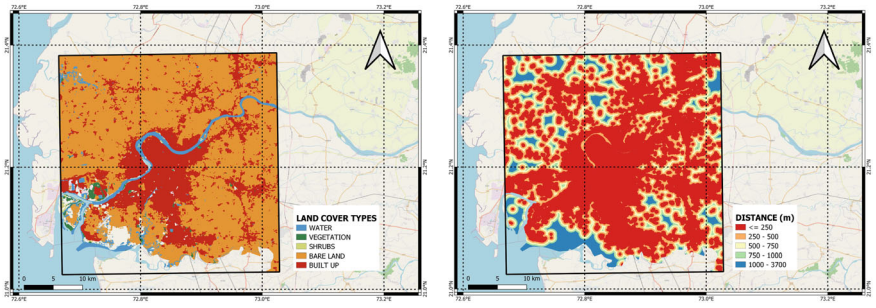
4 Results

Landfill Evaluation Criteria

The thematic raster layers corresponding to the thirteen landfill site selection criteria are represented from Figs. 4, 5, 6, 7, 8, 9 and 10. Figure 4a illustrates the relative suitability for landfill site development based on Surat's regional Land Use and Land Cover (LULC). Bare lands are deemed most suitable, followed by shrubs, vegetation, and built-up areas. It shows that bare lands are the most suitable, followed by shrubs, vegetation, and built-up areas. Bare lands are considered ideal for landfill development due to their sparse vegetation and lack of infrastructure, providing ample space with minimal environmental impact. Shrubs are moderately suitable, requiring some clearing but less than denser vegetation, and they can also act as a buffer. In contrast, built-up areas are the least suitable due to their infrastructure, population density, and environmental regulations, presenting challenges such as environmental impact and community acceptance.

Figure 4b shows the relative suitability of various areas for landfill site development based on their proximity to existing built-up areas, ranging from 0 to 3700 m. The classification emphasizes selecting landfill sites farther away from built-up areas to reduce potential environmental and health impacts. Areas were categorized into five levels based on their proximity to existing built-up spaces: Unsuitable (0–250 m), Less Suitable (250–500 m), Moderately Suitable (500–750 m), Fairly Suitable (750–1000 m), and Most Suitable (>1000 m).

Figure 5a, b illustrate the relative suitability of various regions for landfill site development based on ground elevation and slope, respectively. Lower elevations were preferred for landfill sites due to easier access and potentially lower environmental impacts. The regional elevation ranged from 0 to 3700 m, with areas up to 740 m considered most suitable, 740–1480 m as fairly suitable, and 1480–2220 m as moderately suitable. Elevations from 2220 to 2960 m were considered less suitable, and those above 2960 m were unsuitable for landfill site development. Similarly, gentle slopes were preferred for landfill sites due to stability, ease of construction,



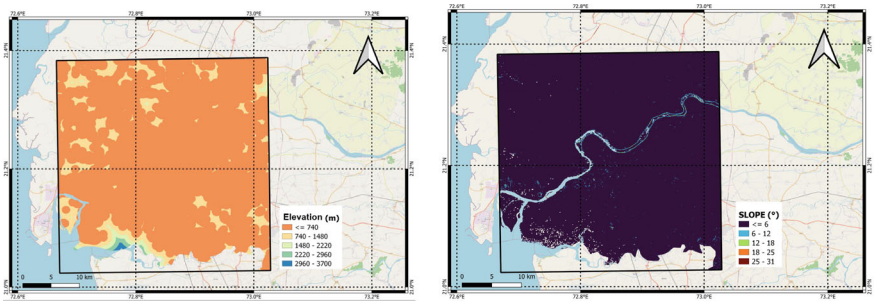
(a) Land Use Land Cover (b) Built Up Proximity

Fig. 4 **a** Land Use Land Cover Map and **b** Built-Up Proximity Map for Surat city

and reduced erosion risks. They offered lower environmental impacts and were easier to manage and maintain. Surat's slope ranged from 0° to 31°, with ground slopes from 0° to 6° classified as most suitable, 6° to 12° as fairly suitable, 12° to 18° as moderately suitable, 18° to 25° as less suitable and greater than 25° as unsuitable for landfill development.

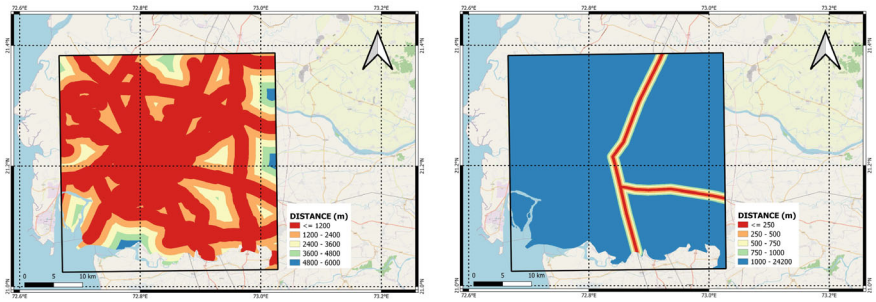
Figure 6a, b depict the relative suitability of areas for landfill site development based on their proximity to roadways and railways, respectively. Proximity to roadways was preferred for landfill sites due to shorter routes and reduced logistical challenges. This proximity could also minimize the need for new infrastructure, such as road construction, which could be costly and time-consuming. The road proximity across Surat ranged from up to 6 km, with areas within 1200 m classified as Most Suitable, 1200–2400 m as Fairly Suitable, 2400–3600 m as Moderately Suitable, 3600–4800 m as Less Suitable, and 4800–6000 m as Unsuitable for landfill site development. Further, Railway proximity in the region varied from 0 m to 24.2 km. Greater distances from railway lines were preferred for landfill sites to minimize risks and avoid conflicts with railway operations. Areas within 250 m of railway lines were deemed unsuitable, 250–500 m were less suitable, 500–750 m were moderately suitable, 750–1000 m were fairly suitable, and distances greater than 1000 m were considered most suitable for landfill site development.

Figure 7a, b depicts the relative suitability of areas for landfill site development based on airport proximity and land value, respectively. Landfill sites were preferred to be located away from airports to reduce impacts on airport operations, conflicts with airport functions, and issues such as foul smells and bird strikes. The regional airport proximity ranged up to 3.94 km. Based on airport proximity, areas within 250 m were classified as Unsuitable, followed by 250–500 m as less suitable, 500–750 m as moderately suitable, 750–1000 m as fairly suitable, and >1000 m as most suitable. Further, as shown in Fig. 6b areas with lower land values were preferred for landfill site development. Selecting sites with lower land values can help minimize costs associated with land acquisition for landfill purposes. Additionally, choosing sites with lower land values may also reduce potential conflicts with other land uses



(a) Ground Elevation (b) Ground Slope

Fig. 5 **a** Ground Elevation Map and **b** Ground Slope Map for Surat city

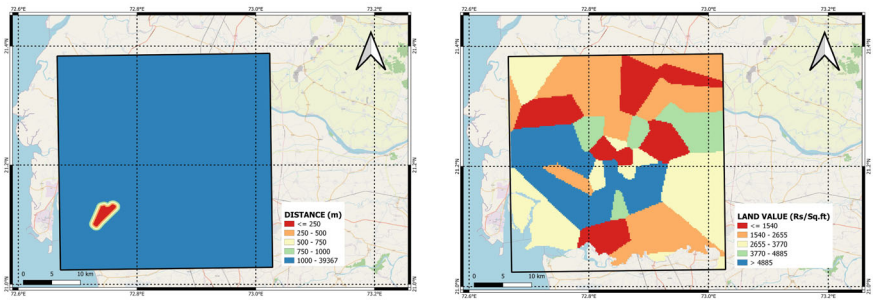


(a) Roadways Proximity (b) Railways Proximity

Fig. 6 **a** Roadways Proximity Map and **b** Railways Proximity Map for Surat city

and development plans, making them more suitable for landfill development. Hence, the Surat region is classified into five suitability levels based on their value: per square foot. from most suitable (<= Rs 1540/sq. ft) to fairly suitable (Rs 1540–2655 /sq. ft), moderately suitable (Rs 2655–3770 /sq. ft), less suitable (Rs 3770–4885 /sq. ft), and unsuitable (> Rs 4885 /sq. ft).

Figure 8a, b illustrates the relative suitability of areas for landfill site development based on their proximity to vegetated areas and water bodies. The selection of these sites away from vegetated areas was preferred to reduce environmental impact and conflicts with natural habitats. The regional vegetation proximity across Surat ranged up to 13.36 km. The locations within 250 m of vegetation coverage were deemed unsuitable, followed by 250–500 m as less suitable, 500–750 m as moderately suitable, 750–1000 m as fairly suitable and greater than 1000 m as most suitable for landfill site development. Further, the regional water-body proximity ranged up to 10.52 kms and landfill sites were preferred away from water-bodies to minimize environmental impacts, avoid water source contamination, and protect aquatic ecosystems and water quality. Hence as shown in Fig. 6b locations within close proximity (<250 m) to water bodies were considered unsuitable, followed by



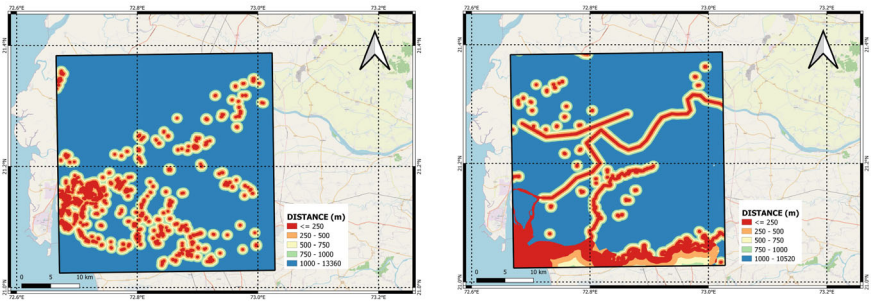
(a) Airport Proximity (b) Land Value

Fig. 7 **a** Airport Proximity Map and **b** Land Value Map for Surat city

250–500 m as less suitable, 500–750 m as moderately suitable, 750–1000 m as fairly suitable and greater than 1000 m as most suitable for landfill site development.

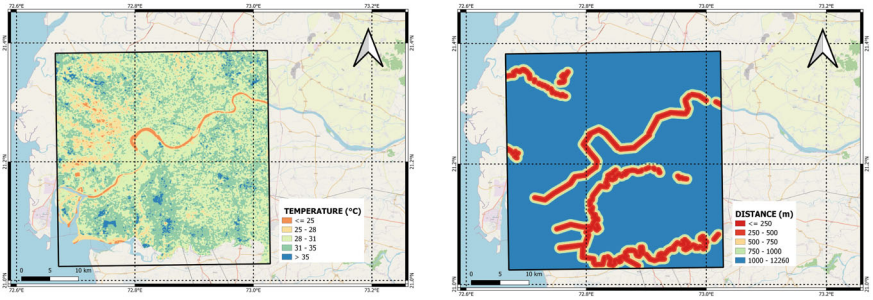
Figure 9a, b illustrate the suitability of areas for landfill site development based on land surface temperatures (LST) and proximity to lineaments, respectively. Higher LSTs indicate lower suitability due to potential challenges such as heat stress, increased emissions, and health risks to workers. Therefore, locations with LSTs lower than 25 °C were deemed most suitable, followed by 25–28 °C as fairly suitable, 28–31 °C as less suitable, and greater than 35 °C as unsuitable for landfill site development. Additionally, greater distances from lineaments were preferred for landfill development to minimize environmental impacts, avoid disturbing geological features, and maintain land stability, thereby reducing the risk of subsidence and other geological hazards. The regional lineament proximity ranged up to 12.26 km, with areas in close vicinity (<250 m) deemed unsuitable, followed by 250–500 m as less suitable, 500–750 m as moderately suitable, 750–1000 m as fairly suitable, and greater than 1000 m as most suitable for landfill development.

Figure 10 displays the relative suitability of different locations in Surat for landfill site development as per their groundwater depth. Shallow groundwater tables were



(a) Vegetation Proximity (b) Waterbody Proximity

Fig. 8 **a** Vegetation Proximity Map and **b** Waterbody Proximity Map for Surat city



(a) Land Surface Temperature (b) Lineament Proximity

Fig. 9 **a** Land Surface Temperature Map and **b** Lineament Proximity Map for Surat city

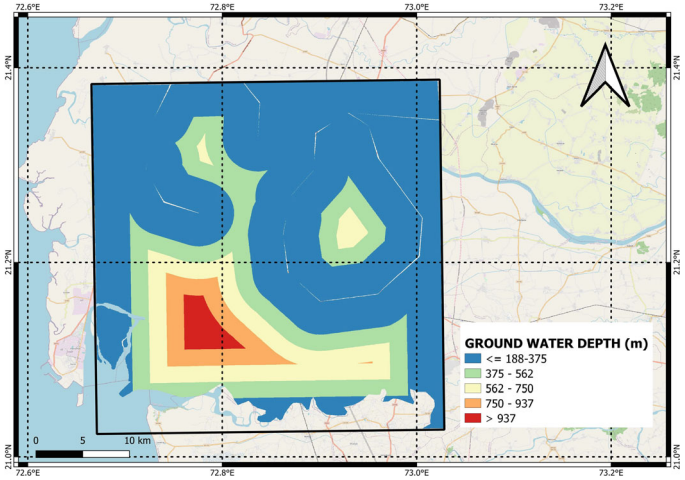


Fig. 10 Ground water depth map for Surat city

preferred for landfill development to reduce the risk of groundwater contamination from leachate. Conversely, areas with deeper groundwater tables are less suitable due to higher contamination risks. The most suitable areas for landfill site development have a low groundwater table depth, ranging from ≤188–375 m, followed by fairly suitable areas at 375–562 m, moderately suitable areas at 562–750 m, less Suitable areas at 750–937 m and unsuitable areas with depths greater than 937 m.

Analytic Hierarchy Process

Table 6 illustrates the pairwise comparison matrix derived from the arithmetic mean of responses from ten subject experts, indicating the relative importance of criteria for landfill site selection. Table 7 displays the normalized pair-wise comparison entries.

Table 8 displays the AHP criteria weights and consistency check results. Criteria weights were calculated by averaging pairwise comparison matrix values row-wise. Column-wise weighted sum values were obtained by multiplying criteria weights with corresponding row-wise values in Table 8. Principal eigenvectors were calculated by dividing the weighted sum values by criteria weights. Lambda max was calculated as the average of principal eigenvectors in Table 8. The Consistency Index value of 0.007 was obtained using Eq. 2, and the Consistency Ratio value of 0.005 was calculated by dividing this value by the randomness index value of 1.56 for 13 evaluation criteria. This ratio, significantly lower than the allowable threshold of 0.10, validated the consistency of the results.

Weighted Overlay Analysis

The top five criteria—LULC at 0.162, BUP at 0.125, WBP at 0.105, RoP at 0.105, and RaP at 0.084—emerged as the most influential for landfill site selection. Conversely,

Table 6 Pairwise comparison matrix for the thirteen landfill evaluation criteria

	LULC	BUP	WBP	RoP	RaP	LV	Ap	VP	LST	GS	LiP	GEl	GWT
LULC	1	1.65	1.6	1.6	2.1	2.3	2.3	3.1	3.1	3.1	3.8	3.2	3.4
BUP	0.61	1	1.08	1.13	1.5	2.1	2.12	2.3	2.3	2.3	3.1	3.1	3.6
WBP	0.63	0.93	1	1.01	1.11	1.2	1.38	2.11	2.11	2.12	2.6	2.62	3.14
RoP	0.63	0.88	0.99	1	1.2	1.4	1.4	2.1	2.1	2.12	2.4	2.4	3.1
RaP	0.48	0.67	0.9	0.83	1	1.3	1.3	1.4	1.4	1.4	2.1	2.1	2.4
LV	0.43	0.48	0.83	0.71	0.77	1	1.05	1.17	1.16	1.18	1.42	1.41	1.62
Ap	0.43	0.47	0.72	0.71	0.77	0.95	1	1.15	1.16	1.17	1.4	1.4	1.6
VP	0.32	0.43	0.47	0.48	0.71	0.85	0.87	1	1.03	1.1	1.9	1.8	1.31
LST	0.32	0.43	0.47	0.48	0.71	0.86	0.86	1.01	1	1.1	1.2	1.2	1.3
GS	0.32	0.43	0.47	0.47	0.71	0.85	0.85	0.91	0.91	1	1.2	1.2	1.3
LiP	0.26	0.32	0.38	0.42	0.48	0.7	0.71	0.53	0.83	0.83	1	1.1	1.2
GEl	0.31	0.32	0.38	0.42	0.48	0.71	0.71	0.56	0.83	0.83	0.91	1	1.9
GWT	0.29	0.28	0.32	0.32	0.42	0.62	0.63	0.76	0.77	0.77	0.83	0.53	1

Table 7 Normalized pair-wise comparison matrix for the thirteen landfill evaluation criteria

	LULC	BUP	WBP	RoP	RaP	LV	Ap	VP	LST	GS	LiP	GEl	GWT
LULC	0.17	0.20	0.17	0.17	0.18	0.15	0.15	0.17	0.17	0.16	0.16	0.14	0.13
BUP	0.10	0.12	0.11	0.12	0.13	0.14	0.14	0.13	0.12	0.12	0.13	0.13	0.13
WBP	0.10	0.11	0.10	0.11	0.09	0.08	0.09	0.12	0.11	0.11	0.11	0.11	0.12
RoP	0.10	0.11	0.10	0.10	0.10	0.09	0.09	0.12	0.11	0.11	0.10	0.10	0.12
RaP	0.08	0.08	0.09	0.09	0.08	0.09	0.09	0.08	0.07	0.07	0.09	0.09	0.09
LV	0.07	0.06	0.09	0.07	0.06	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06
Ap	0.07	0.06	0.07	0.07	0.06	0.06	0.07	0.06	0.06	0.06	0.06	0.06	0.06
VP	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.05	0.06	0.08	0.08	0.05
LST	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.05	0.06	0.05	0.05	0.05
GS	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05
LiP	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.03	0.04	0.04	0.04	0.05	0.04
GEl	0.05	0.04	0.04	0.04	0.04	0.05	0.05	0.03	0.04	0.04	0.04	0.04	0.07
GWT	0.05	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.03	0.02	0.04

GWT at 0.037, GEI at 0.044, LiP at 0.042, GS at 0.052, and GS at 0.052 exhibited relatively lower impact on the selection process. The Landfill Suitability Index (LSI) is computed for the entire study area using reclassified raster layers and their respective criteria weights, as per Eq. 4. LSI values across the study area ranged from 0 to 4.73. These scores were used to classify the area into different suitability classes. Table 9 represents the absolute area and percentage coverage under each suitability class. The Most Suitable class covered 10.89% of the total area, followed

Table 8 AHP criteria weights and consistency check results

Criteria	Weighted sum value	Criteria weights	Principal eigen vectors	Lambda max	CI	CR
LULC	2.324	0.162	13.087	13.084	0.007	0.005
BUP	1.800	0.125	13.052			
WBP	1.513	0.105	13.087			
RoP	1.512	0.105	13.040			
RaP	1.208	0.084	13.036			
LV	0.951	0.066	13.075			
AP	0.928	0.065	13.045			
VP	0.826	0.058	13.163			
LST	0.772	0.054	13.027			
GS	0.753	0.052	13.024			
LiP	0.609	0.042	13.076			
GEl	0.632	0.044	13.209			
GWT	0.531	0.037	13.176			

by the Fairly Suitable class, which accounted for 42.06% of the area. The Moderately Suitable class covered 26.25% of the area, while the Less Suitable class covered 18.23%. The smallest area percentage was classified as Unsuitable, covering 2.57% of the total area.

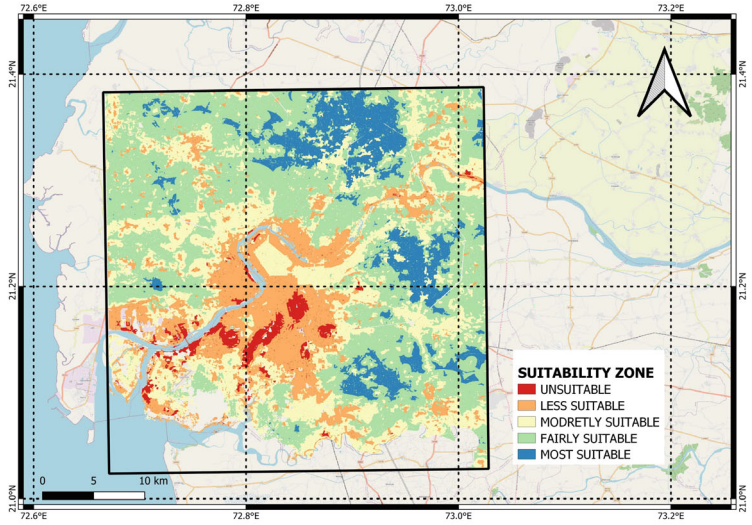
The landfill site suitability map for Surat city is shown in Fig. 11. Fifteen candidate landfill sites are selected across Talangpor (L1), Umber Village (L2, L3), Kamrej (L4), Pasondara (L5), Kamrej (L6, L7, L8), Gothan (L9), Umra (L10), Kathodara (L12), Hazira (L13), Vanz (L14) and Sarol (L15) areas.

TOPSIS Based Landfill Site Ranking

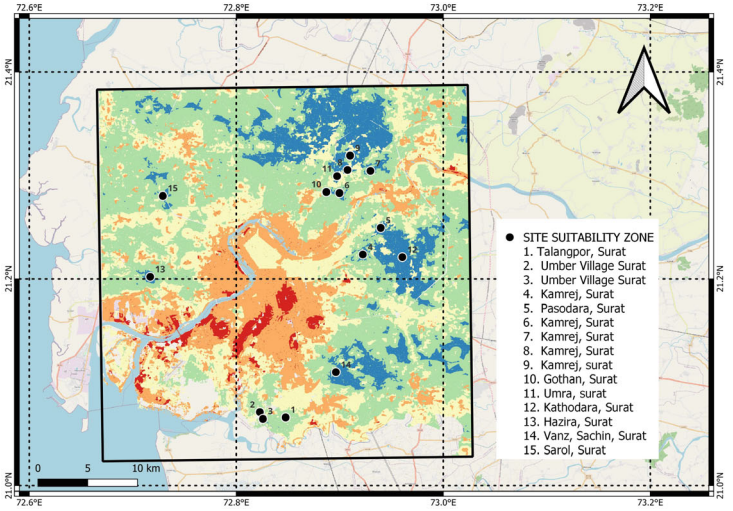
The TOPSIS MCDM method was utilized to determine the optimal landfill site, prioritizing options that are nearest to the positive ideal solution and farthest from the negative ideal solution, based on various evaluation criteria. Table 10 presents

Table 9 Absolute and Percentage Area Coverage under different suitability classes

Suitability class	Unsuitable	Less suitable	Moderate suitable	Fairly suitable	Most suitable	Total
Landfill Suitability Index	<= 0.26	2.62–3.15	3.15–3.67	3.67–4.20	4.20–4.73	
Area in sq. km	32.06	227.82	327.92	525.55	136.04	1250
% Area Coverage	2.57%	18.23%	26.25%	42.06%	10.89%	100%



(a) Landfill Suitability Index Map



(b) Potential Landfill Locations

Fig. 11 **a** Landfill Suitability Index Map **b** Fifteen potential landfill locations around Surat city

the Initial Decision Matrix with criteria-wise values for sixteen candidate locations, while Tables 11 and 12 shows the Normalization and weighted Normalization Matrix to standardize the values on a scale from 0 to 1. TOPSIS results comprising the ideal and anti-ideal solutions for the fifteen landfill sites and three most optimal sites are shown in Fig. 12a, b respectively.

Table 10 Initial Decision Matrix showing criteria values for the fifteen candidate locations

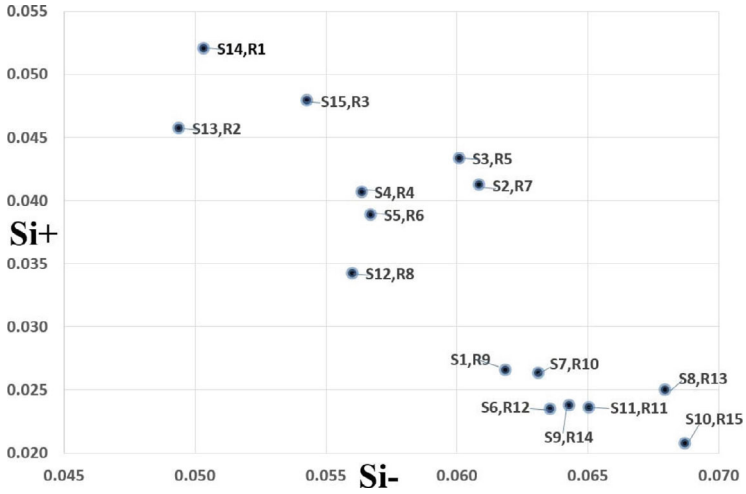
	LULC	BUP	WBP	RoP	RaP	LV	AP	VP	LST	GS	LiP	GEl	GWT
L1	5	376	4305	918	2983	1909	11,389	1370	31	0.93	2340	676	431
L2	5	103	3437	2621	5351	1720	9079	832	26	1.05	2183	1191	491
L3	5	1191	2109	2528	5231	960	9798	1018	26	1.02	2259	1050	419
L4	5	1050	2639	1977	6882	960	19,590	1651	34	1.02	3213	591	605
L5	5	676	2355	729	7608	960	22,667	2106	32	3.11	3291	441	540
L6	5	502	2402	1016	2457	1450	22,159	1534	33	1.71	2080	644	127
L7	5	441	3335	894	4432	1056	25,991	1390	32	0.99	2395	739	314
L8	5	158	6181	320	2302	1056	24,586	2575	33	1.49	3300	136	131
L9	5	136	3360	1179	1901	1720	25,974	1967	31	1.02	4641	321	146
L10	5	591	3271	582	1175	425	21,419	2218	29	1.39	3304	537	18
L11	5	644	2146	791	1618	1720	23,420	2819	31	1.64	3312	103	64
L12	5	739	2426	830	6980	1720	22,930	3904	31	0.74	4650	158	416
L13	5	321	4696	1266	12,837	1720	8518	2268	26	2.48	3485	1067	489
L14	5	1067	4398	618	3093	5939	14,126	1746	29	1.48	5335	376	554
L15	5	537	2410	796	13,586	1720	16,751	7491	25	0.74	4504	502	7

Table 11 Normalization Matrix for the fifteen candidate landfill locations

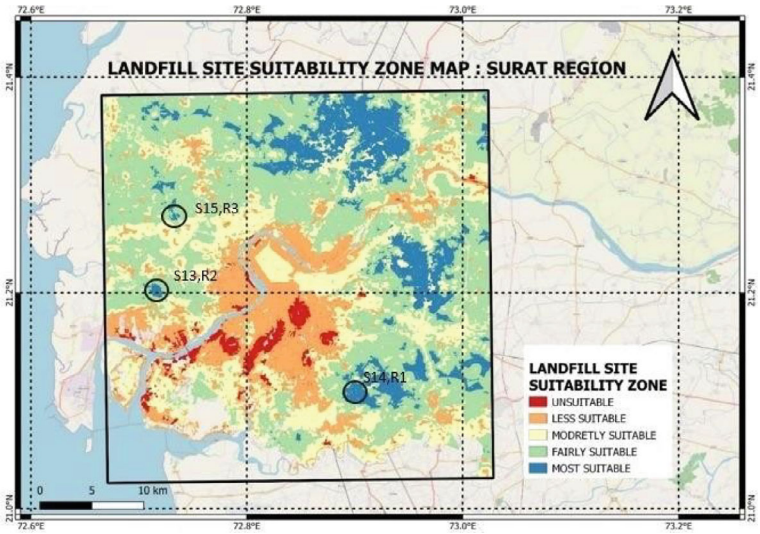
	LULC	BUP	WBP	RoP	RaP	LV	AP	VP	LST	GS	LiP	GEl	GWT
L1	0.26	0.15	0.32	0.18	0.12	0.24	0.15	0.13	0.27	0.16	0.17	0.27	0.29
L2	0.26	0.04	0.25	0.51	0.22	0.22	0.12	0.08	0.22	0.18	0.16	0.47	0.34
L3	0.26	0.47	0.16	0.49	0.21	0.12	0.13	0.09	0.22	0.17	0.17	0.41	0.29
L4	0.26	0.41	0.20	0.39	0.28	0.12	0.26	0.15	0.29	0.17	0.24	0.23	0.41
L5	0.26	0.27	0.17	0.14	0.31	0.12	0.30	0.19	0.28	0.53	0.24	0.17	0.37
L6	0.26	0.20	0.18	0.20	0.10	0.18	0.29	0.14	0.28	0.29	0.15	0.25	0.09
L7	0.26	0.17	0.25	0.17	0.18	0.13	0.34	0.13	0.27	0.17	0.18	0.29	0.21
L8	0.26	0.06	0.46	0.06	0.09	0.13	0.32	0.24	0.29	0.25	0.24	0.05	0.09
L9	0.26	0.05	0.25	0.23	0.08	0.22	0.34	0.18	0.26	0.17	0.34	0.13	0.10
L10	0.26	0.23	0.24	0.11	0.05	0.05	0.28	0.20	0.25	0.24	0.24	0.21	0.01
L11	0.26	0.25	0.16	0.15	0.07	0.22	0.31	0.26	0.27	0.28	0.24	0.04	0.04
L12	0.26	0.29	0.18	0.16	0.28	0.22	0.30	0.36	0.26	0.13	0.34	0.06	0.28
L13	0.26	0.13	0.35	0.25	0.52	0.22	0.11	0.21	0.22	0.42	0.26	0.42	0.33
L14	0.26	0.42	0.33	0.12	0.12	0.74	0.19	0.16	0.25	0.25	0.39	0.15	0.38
L15	0.26	0.21	0.18	0.16	0.55	0.22	0.22	0.69	0.22	0.13	0.33	0.20	0.00

Table 12 Weighted Normalization Matrix for the fifteen candidate landfill locations

	LULC	BUP	WBP	RoP	RaP	LV	Ap	VP	LST	GS	LiP	GEl	GWT
L1	0.014	0.008	0.018	0.010	0.007	0.013	0.008	0.007	0.015	0.009	0.010	0.015	0.016
L2	0.014	0.002	0.014	0.028	0.012	0.012	0.007	0.004	0.012	0.010	0.009	0.026	0.018
L3	0.014	0.026	0.009	0.027	0.012	0.007	0.007	0.005	0.012	0.009	0.009	0.023	0.016
L4	0.014	0.023	0.011	0.021	0.015	0.007	0.014	0.008	0.016	0.009	0.013	0.013	0.023
L5	0.014	0.015	0.010	0.008	0.017	0.007	0.016	0.011	0.015	0.029	0.013	0.010	0.020
L6	0.014	0.011	0.010	0.011	0.005	0.010	0.016	0.008	0.015	0.016	0.008	0.014	0.005
L7	0.014	0.010	0.014	0.010	0.010	0.007	0.019	0.007	0.015	0.009	0.010	0.016	0.012
L8	0.014	0.003	0.025	0.003	0.005	0.007	0.018	0.013	0.016	0.014	0.013	0.003	0.005
L9	0.014	0.003	0.014	0.013	0.004	0.012	0.019	0.010	0.014	0.009	0.019	0.007	0.005
L10	0.014	0.013	0.013	0.006	0.003	0.003	0.016	0.011	0.014	0.013	0.013	0.012	0.001
L11	0.014	0.014	0.009	0.009	0.004	0.012	0.017	0.014	0.015	0.015	0.013	0.002	0.002
L12	0.014	0.016	0.010	0.009	0.015	0.012	0.017	0.020	0.015	0.007	0.019	0.003	0.016
L13	0.014	0.007	0.019	0.014	0.028	0.012	0.006	0.011	0.012	0.023	0.014	0.023	0.018
L14	0.014	0.023	0.018	0.007	0.007	0.041	0.010	0.009	0.014	0.014	0.022	0.008	0.021
L15	0.014	0.012	0.010	0.009	0.030	0.012	0.012	0.038	0.012	0.007	0.018	0.011	0.000



(a) TOPSIS Ideal and Anti Ideal Solutions



(b) Optimal Landfill Site Locations

Fig. 12 **a** TOPSIS Results and **b** Optimal Landfill locations across Surat

5 Conclusions

The rapid overhaul of existing urban landfills requires scientific approaches to select new municipal landfill sites. An integrated Geographic Information System (GIS)–Multi Criteria Decision-Making (MCDM) framework was presented encompassing

thirteen Evaluation criteria (including Land Use Land Cover, Built-Up Proximity, Ground Elevation, Ground Slope, Road Proximity, Rail Proximity, Airport Proximity, Land Value, Vegetation Proximity, Water Body Proximity, Land Surface Temperature, Lineament Proximity, Ground Water Depth) to propose new municipal landfill sites for Surat city in the western Indian state of Gujarat, India. Thematic raster layers for these layers were reclassified on a 1–5 scale. Based on inputs from 10 subject experts, Analytical Hierarchy Process (AHP) was employed to ascertain the relative contribution of chosen landfill criteria. Further, weighted overlay analysis is applied to develop an overall suitability map. Results showed that out of the total 1250 sq. km area, 10.89% was highly suitable, 42.06% was fairly suitable, 26.25% was moderately suitable, 18.23% was less suitable, and 2.57% was unsuitable for landfill development.

Further, TOPIS MCDM technique was utilized to rank 15 potential landfill sites within the most suitable areas, with Vanz, Sachin, Sarol, and Kamrez emerging as the top three choices for future development as municipal landfill sites. While other locations were technically feasible, they were considered less optimal for landfill site development. This research offered valuable guidance for urban planners and policymakers, facilitating a shift from traditional to evidence-based approaches in landfill site selection. This study provided valuable guidance for urban planners and policymakers, encouraging a shift from traditional to evidence-based approaches in landfill site selection. However, limitations, such as data availability and accuracy, as well as stakeholder engagement, could affect the model's effectiveness. Future research could integrate additional relevant landfill selection criteria and also conduct sensitivity analyses to assess the impact of criteria weightages on the final zoning and ranking of landfill sites.

References

1. Ajim, S., Ateeque, A.: Suitability analysis for municipal landfill site selection using fuzzy analytic hierarchy process and geospatial technique. Environ. Earth Sci. (2020).https://doi.org/10.1007/s12665-020-08970-z
2. Ali, S.A., Parvin, F., Al-ansari, N., Pham, Q.B., Ahmad, A., Raj, M.S.: Sanitary landfill site selection by integrating AHP and FTOPSIS with GIS: a case study of Memari Municipality. Environmental Science and Pollution Research, India (2020). https://doi.org/10.1007/s11356-020-11004-7
3. Balew, A., Alemu, M., Leul, Y., Feye, T.: Suitable landfill site selection using GIS-based multi-criteria decision analysis and evaluation in Robe town, Ethiopia. GeoJournal, 3 (2020).https://doi.org/10.1007/s10708-020-10284-3
4. Barzehkar, M., Dinan, N.M., Mazaheri, S., Tayebi, R.M., Brodie, G.I.: Landfill site selection using GIS-based multi-criteria evaluation (case study: SaharKhiz Region located in Gilan Province in Iran). SN Appl. Sci. **1**(9) (2019). https://doi.org/10.1007/s42452-019-1109-9
5. CPHEEO: Central Public Health and Environmental Engineering Organization. Manual on Municipal Solid Waste Management, First Edition (2016)
6. Ding, Z., Zhu, M., Wang, Y., Zhu, J.: An AHP-GIS Based Model of C & D Waste Land fill Site Selection: A Triangulation of Critical Factors, 163–174 (2018)

7. Dolui, S., Sarkar, S.: Identifying potential landfill sites using multicriteria evaluation modeling and GIS techniques for Kharagpur city of West Bengal. India. Environ. Challenges **5**, 100243 (2021). https://doi.org/10.1016/j.envc.2021.100243
8. Elkhrachy, I., Alhamami, A., Alyami, S.H.: Landfill site selection using multi-criteria decision analysis, remote sensing data, and geographic information system tools in Najran City, Saudi Arabia. Remote Sensing **15**(15) (2023). https://doi.org/10.3390/rs15153754
9. Hazarika, R., Saikia, A.: Landfill site suitability analysis using AHP for solid waste management in the Guwahati Metropolitan Area, India. Arabian J. Geosci. (2020)
10. Islam, A., Sanzida Murshed, M.H.: Selecting suitable landfill site with multi-criteria evaluation and GIS : a case of Savar upazila in Bangladesh. Arabian J. Geosciences **2** (2020).https://doi.org/10.1007/s12517-020-05925-3
11. Jamshidi-Zanjani, A., Rezaei, M.: Landfill site selection using combination of fuzzy logic and multi-attribute decision-making approach. Environ. Earth Sci. **76**(13) (2017). https://doi.org/10.1007/s12665-017-6774-7
12. Kazuva, E., Zhang, J., Tong, Z., Liu, X.P., Memon, S., Mhache, E.: GIS- and MCD-based suitability assessment for optimized location of solid waste landfills in Dar es Salaam. Tanzania. Environ. Sci. Pollut. Res. **28**(9), 11259–11278 (2021). https://doi.org/10.1007/s11356-020-11213-0
13. Monzambe, G.M., Mpofu, K., Daniyan, I.A.: Optimal location of landfills and transfer stations for municipal solid waste in developing countries using non-linear programming. Sustain. Futures **3**, 100046 (2020). https://doi.org/10.1016/j.sftr.2021.100046
14. Mukhalad Muayad, H.J.: Geographic information system (GIS) for Landfill site selection—a review. IOP Conference Series: Earth and Environmental Science (2023).https://doi.org/10.1088/1755-1315/1232/1/012004
15. Nitheshnirmal S., Abdul Rahaman Sheik Mohideen, B.A.: Optimisation of landfill sites for solid waste disposal in Thiruverumbur taluk of Tiruchirappalli district, India. Environ. Earth Sci. Springer (2020).https://doi.org/10.1007/s12665-020-09264-0
16. Nyimbili, P.H., Erden, T.: Socio-economic planning sciences GIS-based fuzzy multi-criteria approach for optimal site selection of fire stations in Istanbul. Turkey. Socio-Econo. Planning Sci. **71**, 100860 (2020). https://doi.org/10.1016/j.seps.2020.100860
17. Özkan, B., Özceylan, E., Sarıçiçek, İ.: GIS-based MCDM modeling for landfill site suitability analysis: a comprehensive review of the literature. Environ. Sci. Pollut. Res. **26**(30), 30711–30730 (2019). https://doi.org/10.1007/s11356-019-06298-1
18. Patil, R.B.: An infrastructural survey on biomedical waste management in Nashik City. J. Environ. Impact Manag. Policy **2**(3), 1–18 (2022). https://doi.org/10.55529/jeimp23.1.18
19. Paul, S., Ghosh, S.: Cleaner logistics and supply chain identi fi cation of solid waste dumping site suitability of Kolkata Metropolitan Area using Fuzzy-AHP model. Cleaner Logist. Supply Chain **3**, 100030 (2022). https://doi.org/10.1016/j.clscn.2022.100030
20. Rahimi, S., Hafezalkotob, A., Monavari, S.M., Hafezalkotob, A., Rahimi, R.: Sustainable landfill site selection for municipal solid waste based on a hybrid decision-making approach: fuzzy group BWM-MULTIMOORA-GIS. J. Cleaner Prod., 119186 (2019).https://doi.org/10.1016/j.jclepro.2019.119186
21. Roy, D., Das, S., Paul, S., Paul, S.: An assessment of suitable landfill site selection for municipal solid waste management by GIS—based MCDA technique in Siliguri municipal corporation planning area.. Comput. Urban Sci. (2022).https://doi.org/10.1007/s43762-022-00038-x
22. Saaty, R.W.: The analytic hierarchy process-what it is and how it is used. Math. Model. **9**(3–5), 161–176 (1987). https://doi.org/10.1016/0270-0255(87)90473-8
23. Sadhasivam, N., Rahaman, A., Mohideen, S., Alankar, B.: Optimisation of landfill sites for solid waste disposal in Thiruverumbur taluk of Tiruchirappalli district. India. Environ. Earth Sci. **79**(23), 1–20 (2020). https://doi.org/10.1007/s12665-020-09264-0
24. Saha, A., Roy, R.: An integrated approach to identify suitable areas for built-up development using GIS-based multi-criteria analysis and AHP in Siliguri planning area, India. SN Appl. Sci. **3**(4) (2021). https://doi.org/10.1007/s42452-021-04354-5

25. Sk, M., Ajim, S., Ateeque, A.: Optimal sanitary landfill site selection for solid waste disposal in Durgapur City using geographic information system and multi—criteria evaluation technique. KN—J. Cartography Geogr. Inf. **70**(4), 163–180 (2020). https://doi.org/10.1007/s42489-020-00052-1
26. Tercan, E., Ali, M., Serkan, D.: A GIS—based multi - criteria evaluation for MSW landfill site selection in Antalya , Burdur , Isparta planning zone in Turkey. Environ. Earth Sci., 1–17 (2020).https://doi.org/10.1007/s12665-020-08974-9
27. Tripathi, A.K., Agrawal, S., Gupta, R.D.: Comparison of GIS-based AHP and fuzzy AHP methods for hospital site selection : a case study for Prayagraj City. GeoJournal **87**(5), 3507–3528 (2022). https://doi.org/10.1007/s10708-021-10445-y
28. Waheeb, S.A., Zerouali, B., Elbeltagi, A., Alwetaishi, M., Wong, Y.J., Bailek, N., AlSaggaf, A.A., Abd Elrahman, S.I.M., Santos, C.A.G., Majrashi, A.A.: Enhancing sustainable urban planning through GIS and multiple-criteria decision analysis: a case study of green space infrastructure in Taif Province, Saudi Arabia. Water (Switzerland) **15**(17) (2023). https://doi.org/10.3390/w15173031
29. Zambrano-asanza, S., Quiros-tortos, J., Franco, J.F.: Optimal site selection for photovoltaic power plants using a GIS-based multi-criteria decision making and spatial overlay with electric load Nomenclature. Renew. Sustain. Energy Rev. **143**, 110853 (2021). https://doi.org/10.1016/j.rser.2021.110853
30. Zolfaghary, P., Zakerinia, M., Kazemi, H.: A model for the use of urban treated wastewater in agriculture using multiple criteria decision making (MCDM) and geographic information system (GIS). Agric. Water Manag., **243** (2021). https://doi.org/10.1016/j.agwat.2020.106490

Comparing Classification Algorithms for Predicting Spatial Land Cover via Landscape Indices in Nashik, India

Kratika Sharma, Ritu Sharma, and Arun Kumar Wadhwani

Abstract The application of robust machine learning (ML) classification algorithms to connect regional land cover (LC) types such as built-up areas, bare lands, vegetation, and water bodies with landscape indices like Normalized Difference Built-up Index (NDBI), Normalized Difference Bareness Index (NDBaI), Normalized Difference Vegetation Index (NDVI), and Normalized Difference Water Index (NDWI) can facilitate the automatic generation of accurate Land Cover Maps. In this study, eight robust ML algorithms were applied, namely Logistic Regression, Support Vector Regression, Decision Trees, Random Forests, Gaussian Naïve Bayes, Gradient Boosting Classifier, K-Nearest Neighbours, and Artificial Neural Network, to establish a predictive relationship between various LC classes and NDBI, NDBaI, NDVI, and NDWI. LANDSAT satellite datasets were used to provide ample training and testing data for the eight classifiers. Hyperparameter tuning and tenfold cross-validation were employed to achieve the highest possible accuracy. Among the eight classifiers analyzed, the Artificial Neural Network stood out, initially achieving 71.23% accuracy in 2.67 min, matching the highest attainable accuracy after 54.78 min of hyperparameter tuning. The Support Vector Classifier started at 69.69%, reaching 70.00% after 2 h and 49 min. Random Forest, starting at 67.85%, notably increased to 70.26% in a 5.58-h tuning period. Logistic Regression, with an initial 68.57%, improved to 68.76% in just 48 s of tuning.

Keywords Land cover classification · Machine learning · Artificial neural network · Hyper parameter tuning

K. Sharma (✉) · R. Sharma
Computer Science and Engineering, Indian Institute of Information Technology, Pune, India
e-mail: kratikasharma@iiitp.ac.in

A. K. Wadhwani
Electrical Engineering, Madhav Institute of Technology and Science, Gwalior, India

G. Bekdaş and S. M. Nigdeli (eds.), *New Advances in Soft Computing in Civil Engineering*, Studies in Systems, Decision and Control 547,
https://doi.org/10.1007/978-3-031-65976-8_22

1 Introduction

Remote Sensing and Geographic Information Systems (RS-GIS) play a crucial role in urban studies, providing valuable, up-to-date information about urbanized and non-urbanized areas. This information enables informed decision-making and sustainable development [3, 43]. One of the most useful applications of RS-GIS in urban planning is Land Cover (LC) mapping. By utilizing satellite imagery, planners can precisely classify and map different land cover types, such as residential areas, commercial zones, industrial regions, green spaces, and water bodies [23]. This information aids in understanding the existing urban layout and identifying areas for potential expansion or redevelopment. It also facilitates the assessment of urban sprawl and encroachment on natural habitats, thereby helping to preserve ecologically sensitive areas [44]. RS-GIS supports urban risk assessment and disaster management by estimating damage during events like floods, earthquakes, or wildfires, aiding in relief planning [10, 44]. It also analyzes infrared thermal data to identify Urban Heat Islands, enabling strategies for heat mitigation in urban areas [36, 47].

LANDSAT datasets are vital for land cover change assessment and future prediction due to their high spatial and temporal resolution. The LANDSAT program, jointly operated by the National Aeronautics and Space Administration (NASA) and the United States Geological Survey (USGS), provides free mid-resolution datasets, facilitating periodic assessments of spatiotemporal changes [38]. LANDSAT's multi-decadal archive of satellite imagery enables researchers to study long-term trends in land cover changes [2, 33]. Land cover (LC) classification provides valuable information about the distribution and spatial patterns of different land cover types in a given area. However, developing LC maps is a challenging and time-consuming task, especially for large regions with diverse landscapes [20, 26, 37]. This process involves data acquisition from satellite imagery, pre-processing to correct for atmospheric and radiometric effects, and feature extraction to identify relevant spectral and spatial characteristics [42]. One of the critical challenges in LC classification is the subjectivity involved in defining signature classes for each LC class. Signature classes are representative examples of each land cover type used to train the classification algorithms. This step requires expertise and manual processing to select training samples that accurately represent the different LC types in the region [5, 12, 14, 17].

Moreover, LC patterns evolve over time due to seasonal variations or human interventions, requiring regular updates to the classification process. To address these challenges and improve the efficiency and accuracy of LC classification, there is a growing focus on automatic classification methods. Machine learning (ML) algorithms, such as random forests, support vector machines, and artificial neural networks, have shown promising results in automatically classifying land cover types from RS datasets [16, 35]. These algorithms can learn from patterns and relationships in the data, reducing human bias and providing consistent and reproducible classification results. Once trained on a representative dataset, the ML algorithms

can be applied to new areas or future imagery without the need for manual intervention, allowing for faster updates to the LC maps, ensuring they remain relevant and up-to-date.

Predictive classification models that link Land Use with landscape indicators such as NDBI, NDBaI, NDVI, and NDWI through appropriate machine learning (ML) classification algorithms offer a viable option for automated LC mapping [15, 19, 40]. ML algorithms can be trained on previously classified remote sensing (RS) datasets where LC classes are already identified through manual interpretation or ground truth data. Once trained, these algorithms can extrapolate to predict the LC classes for unseen data points based on their quantitative relationships with landscape indicators [13, 48]. Gradually, these ML techniques can adapt and improve with more data, leading to enhanced accuracy and reliability over time. The extraction of NDBI, NDBaI, NDVI, and NDWI values from LANDSAT datasets through distinct band combinations ensures an objective and non-subjective retrieval process. This study employs LANDSAT 8 satellite imagery for the Nashik region, India, to develop a predictive relationship between regional land cover and landscape indicators. Several robust ML classification algorithms are trained on LULC and NDBI, NDBaI, NDVI, and NDWI indices to predict LC classes for unseen data points. These classifiers are evaluated based on their predictive accuracy, highlighting their relative strengths and weaknesses.

2 Literature Review

Recent literature has utilized various classification methods, both unsupervised and supervised, to create regional land cover (LC) maps. Unsupervised classification has employed clustering algorithms like K-means or ISODATA to group pixels based on spectral similarities. For instance, Rahman [42] examined land cover changes in Al-Khobar city, Saudi Arabia, from 1990 to 2013 using the ISODATA unsupervised algorithm with LANDSAT data. The study revealed a notable 117% growth in urban areas from 1990 to 2001, followed by a 43.51% increase from 2001 to 2013, alongside observable vegetation growth. Unsupervised classification methods like K-means or ISODATA group pixels based on spectral similarities, offering a data-driven approach but may not accurately capture specific land cover types. In contrast, supervised methods like MLC, SVM, and Random Forest require labeled data for precise categorization, offering higher accuracy but demanding significant resources and being computationally intensive [5]. Generally, supervised classifiers such as Maximum Likelihood (MLC), Support Vector Machines, and Random Forest deliver higher accuracy because they utilize prior knowledge from training samples [7]. Hussain, Mubeen, and Karuppannan [23] used MLC to examine changes in Pakistan's Okara district, emphasizing its importance in mapping shifts due to population growth. Fahad et al.'s study [16] in Lahore, Pakistan, showcased the MLC algorithm's role in accurately tracking urban expansion. However, certain limitations exist with the use of these approaches. Unsupervised methods lack predefined

categories and may not capture specific land cover types accurately. The supervised LC classification relies on labeled data, demanding substantial resources for training. Both approaches are sensitive to data quality and can be computationally intensive for large-scale mapping [16, 26, 35].

Automated LC Classification

Several studies in literature also developed automated approaches for LC classification to mitigate the constraints of manual classification methods [11]. For instance, Zha et al. [51] introduced an automated method for mapping urban and peri-urban built-up areas. Using NDVI and NDBI from LANDSAT TM data, authors achieved 92.6% mapping accuracy in Nanjing, China, demonstrating NDBI's efficiency and objectivity compared to traditional methods. Li et al. [31] introduced an automatic urban LULC classification approach using LANDSAT-8 OLI sensor data. It employed VWMI, NDVI, and NDBLI to obtain preliminary classifications, enhancing them through a linear model. The method proved effective for urban land-cover mapping. Parent et al. [39] presented a fully automated rule-based algorithm for high-resolution LC classification using LiDAR data and multispectral imagery at a 1-m resolution. S. M. Aswatha et al. [8] proposed a spectral-slope-based automated classification technique for LANDSAT imagery, initially classifying it into water, vegetation, and vegetation-void. They further categorized these into sub-classes through support vector machines and k-means clustering, enabling change quantization in temporal images. Hu et. al [21] adopted Google Earth Engine, classification and regression tree (CART), CVAPS, and NDVI timing analysis for accurate LC map updates in global change research and regional governance. Khalmurzayeva [27] presented a semi-automatic methodology for extracting information from multispectral images, using Sentinel-2a data, including Land Cover Classification System (LCCS) classification, urban and industrial area DEM extraction, and build-up indices.

Role of Landscape Indices

Several landscape indices have also been integrated to enhance automated LC classification and extract key ground features [1, 9, 25, 30, 50]. For example, Ao and Xu [6] employed the Built-Up Index (IBI), derived from SAVI, MNDWI, and NDBI, to efficiently extract urban land-use data in Jinghui irrigation district from remote sensing images across three time periods. Piyoosh and Ghosh [41] focused on Anthropogenic Impervious Surfaces (AIS) extraction using the Normalized Difference Impervious Surface Index (NDAISI), an enhancement of BCI with MNDSI. Szabó et al. [46] investigated the utility of three landscape indices (NDVI, NDWI, MNDWI) from LANDSAT-7 ETM + data for different land cover types. MNDWI, particularly effective for water features, stands out. Firozjaei et al. [28] presented the Automated Built-up Extraction Index (ABEI) for enhanced extraction of built-up areas from LANDSAT 8 OLI data, focusing on challenging environments. ABEI, optimized using the Improved Gravitational Search Algorithm, outperformed UI and

NDBI in accuracy and demonstrated stable threshold values. This makes it effective for various environmental conditions, including bare and sandy areas. Krishnaveni and Anilkumar [28] introduced an innovative automated method for extracting water, built-up, and vegetation land cover datasets using Index Extracted Water Index (IEWI), Index Extracted Built-up Index (IEBI), and Index Extracted Vegetation Index (IEVI).

Landscape indices have also contributed to establishing relationships between LULC and LST [4, 18, 22, 32].

Madasa et al. [34] quantified land-use/cover changes in South Africa's Welkom–Virginia Goldfields, a region heavily impacted by mining. LANDSAT images from 1988 to 2018 were analyzed using geospatial indices, including GEMI, NDBI, NDVI, NDSI, and NDWI, to classify various land cover types. Further, Singh et al. [45] investigated the impact of LU changes on urbanization and its effect on LST in Bhopal, Central India, spanning from 2008 to 2020. Using multi-temporal data, it examines LULC changes and Urban Heat Island UHI, revealing a positive correlation between LST and NDBI, and a negative association with NDVI, indicated by the Pearson coefficient.

In summary, both supervised and unsupervised LC classification approaches pose challenges related to consistency and fatigue, leading to errors and reduced accuracy [24, 49]. In comparison, automated approaches are better suited for large-scale LC mapping requiring fewer human inputs. Further, several studies established predictive relationships between LULC classes and landscape indices [29, 52]. Leveraging historical LC data enables the establishment of statistical relationships between landscape indicators and LULC categories. This, in turn, can facilitate the automated LULC classification of large, uncharted regions. To this end, this research compares the potential of eight widely used classification algorithms for developing a predictive relationship between LC classes and landscape indices.

3 Research Methodology

Study Area

As shown in Fig. 1, Nashik City in Maharashtra, India, situated along the Godavari River, features diverse topography with hills, valleys, and plateaus, including landmarks like Brahmagiri Hill and the Trimbakeshwar plateau. The city experiences a hot semi-arid climate with scorching summers around 40 °C and reduced rainfall due to its rain shadow location, averaging approximately 700 mm annually. Winters bring cooler temperatures of 8–10 °C. This study covers a 337 sq. km area around Nashik to provide sufficient training and testing dataset for the different classifiers. As depicted in Fig. 2, the entire methodology comprises three main steps.

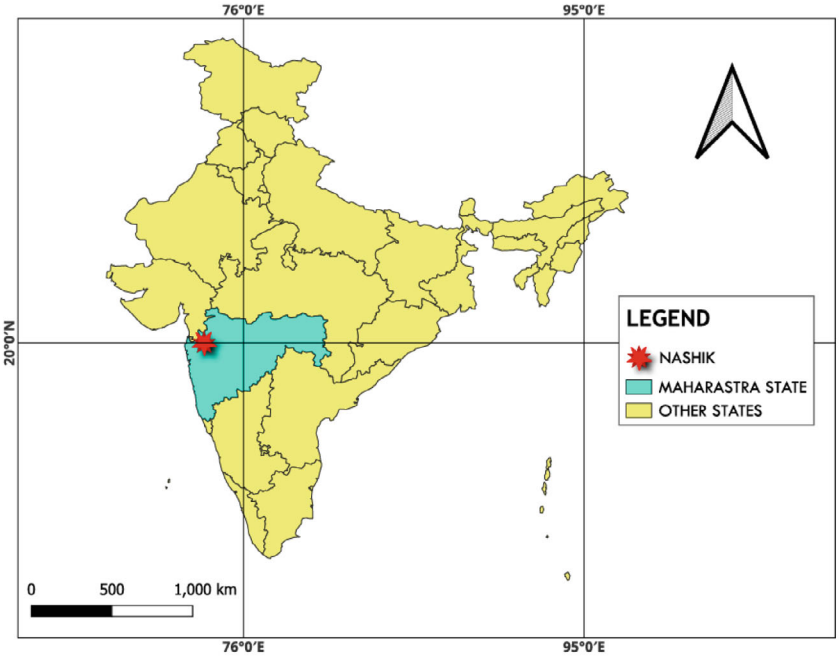


Fig. 1 **a** Geographical location of Nashik city in India Map; **b** Study area chosen for analysis

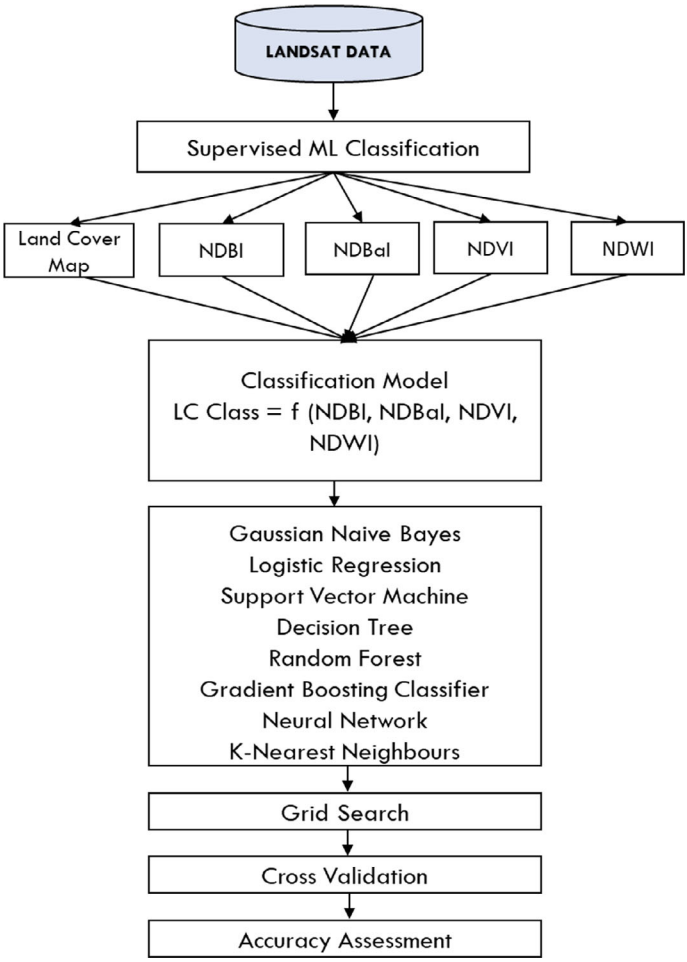


Fig. 2 Stepwise research methodology adopted for this study

Step 1: Develop Land Cover Map

The LANDSAT dataset for the Nashik region on May 12, 2022, was acquired from the USGS Earth Explorer repository. Next, the downloaded image was georeferenced to the Universal Transverse Mercator projection system. Using the Quantum-GIS software, the Maximum Likelihood Classifier (MLC) algorithm was applied to develop LC maps for the Nashik region during May 2022. MLC assigned class labels to pixels based on their spectral characteristics and known class labels from training samples, assuming specific statistical distributions. The process involved data preparation, maximum likelihood estimation, and classification. Training samples were collected for each class in the data preparation phase. The algorithm estimated the mean vector (μ) and covariance matrix for each class using a multivariate Gaussian distribution

assumption in the parameter estimation step.

$$\text{Mean vector } (\mu) \text{ for class k} : \mu\_\text{k} = \frac{1}{n_k} \sum_{i=1}^{n_k} X_i \tag{1}$$

$$\text{Covariance matrix } (\Sigma) \text{for class k} : \Sigma\_\text{k} = \frac{1}{n_k} \sum_{i=1}^{n_k} (X_i - \mu_k).(X_i - \mu_k)^T \tag{2}$$

where n_k is the number of training samples for class k and X_i represent the Spectral feature vector of the i-th training sample. Further, T represents the transpose operation.

ML evaluates the likelihood (L_k) of each pixel's association with each class in the final classification phase, using the estimated parameters and the multivariate Gaussian distribution equation for pixels with spectral feature vector X belonging to class k.

$$\text{Likelihood } \mathrm{L_{k(x)}} = \frac{1}{\left((2\pi)^{\frac{d}{2}}.(|\Sigma_\mathrm{k}|)^{\frac{1}{2}}\right).e^{(-0.5(X_i-\mu_k)^T.\Sigma_\mathrm{k}{}^{-1}.(X_i-\mu_k))}} \tag{3}$$

where d represents the dimensionality of the spectral feature vector, $|\Sigma_\mathrm{k}|$ indicates the determinant of the covariance matrix Σ_k, $(X_i - \mu_k)$ represents the difference between the spectral feature vector X and the mean vector μ_k and finally $\Sigma_\mathrm{k}{}^{-1}$ represents the inverse of the covariance matrix Σ_k. MLC assigned pixels to the class with the highest likelihood, assuming class independence and a normal distribution without considering spatial context. As illustrated in Table 1, the entire study region was categorized into four primary Land Cover (LC) classes: Built-up, Vegetation, Water body, and Bareland. Built-up areas included roads and buildings, while the Water body class comprised rivers and lakes. The Vegetation class included trees, forests, gardens, and cropped farmlands, while Barelands included open areas, uncropped agricultural lands, and landfills.

Next, Kappa statistics (KS) were computed to ascertain the satisfactory classification accuracy of the 2022 LC map. KS employs Cohen's kappa coefficient for this evaluation. This study adopted one hundred twenty ground control points for

Table 1 The different land features considered in four Land Cover classes

Land cover class	Description
Waterbody	Rivers, Lakes, and Wetlands
Vegetation	Trees, Forests, Gardens, and Cultivated Farmlands
Built-Up	Residential, Commercial, Industrial, and Institutional Buildings
Bareland	Bare plots, uncultivated agricultural lands, Landfills and other land cover types

comparison, with a minimum kappa accuracy threshold of 0.80 to accept the ML classification results.

Step 2: Compute LC and Landscape Indices

This stage involved calculating the landscape indices for the entire region using the appropriate frequency bands from the multi-spectral LANDSAT dataset. NDBI detects urban areas based on Near-Infrared (NIR) and Shortwave Infrared (SWIR) reflectance differences. NDBaI assesses bare land using SWIR and red bands. NDVI measures vegetation density with NIR and red differences. NDWI identifies water bodies using green and NIR differences.

$$\text{NDBI} = (\text{NIR} - \text{SWIR})/(\text{NIR} + \text{SWIR}) \tag{4}$$

$$\text{NDBaI} = (\text{SWIR} - \text{Red})/(\text{SWIR} + \text{Red}) \tag{5}$$

$$\text{NDVI} = (\text{NIR} - \text{Red})/(\text{NIR} + \text{Red}) \tag{6}$$

$$\text{NDWI} = (\text{SWIR} - \text{NIR})/(\text{SWIR} + \text{NIR}) \tag{7}$$

Post landscape indices computation, the study area was partitioned into a (50 × 50) m resolution grid to compute the NDBI, NDBaI, NDVI, NDWI, and LC values. These values were structured into a tabular dataset comprising 13,444 rows and 5 columns. The dataset was analyzed in Jupyter Notebook using Python libraries such as Scikit-learn, Keras, Statsmodels, and SciPy, which offer versatile tools for data processing, machine learning, statistics, and scientific analysis, enabling the extraction of valuable insights. The dataset was divided into training (70%) and testing (30%) sets, and eight different classification algorithms, namely Gaussian Naive Bayes, Logistic Regression, Support Vector Machine, Decision Tree, Gradient Boosting Classifier, K-Nearest Neighbours, Random Forest, and Artificial Neural Network, were trained on the training set to predict LC labels for the testing set.

Step 3: Apply Classification Algorithms

Classification algorithms are supervised ML techniques that group data into predefined categories based on labeled training data. These algorithms can identify patterns in input datasets to predict the classes of unseen data. The theoretical foundations and working principles of all eight classification algorithms adopted for this study are presented below.

Logistic Regression

Logistic regression employs a binary classification model that predicts the probability (P) of an instance belonging to class 1 using the logistic function: P(y = 1|X) = 1/(1 + e^(−z)), where 'z' is the linear combination of features (X) and coefficients (β). The model estimates coefficients (β) maximizing the likelihood of the data,

often using optimization algorithms like maximum likelihood or gradient descent. For classification, a decision threshold (usually 0.5) is used. If P(y = 1|X) > 0.5, it's class 1; otherwise, it's class 0. Logistic regression handles multi-class problems using techniques like one-vs-rest or softmax regression.

Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem. It calculates the posterior probability of each class given the features using the formula: P(C|x) = (P(C) * P(x|C))/P(x). The classifier assumes that features are conditionally independent, allowing the likelihood to be represented as a product of individual feature probabilities: P(x|C) = P(×1|C) * P(×2|C) * … * P(xn|C). Parameters such as mean (μ) and variance (σ^2) are estimated for each feature and class from the training data to make predictions for new instances. The class with the highest posterior probability is assigned as the predicted class for the instance.

Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm for binary classification with 'm' instances and 'n' features. It seeks to find the optimal hyperplane represented by the decision function: f(x) = sign(w·x + b), where 'w' is the weight vector, 'x' is the feature vector, and 'b' is the bias term. The goal is to minimize ||w||^2 while satisfying the constraint y_i * (w·x_i + b) ≥ 1 for all instances (x_i, y_i), where y_i is the class label. SVM uses the kernel trick to handle non-linearly separable data, applying a kernel function to implicitly map feature vectors to a higher-dimensional space.

Decision Tree

The decision tree classifier is a supervised learning algorithm for classification and regression tasks. The decision tree classifier recursively partitions instances based on selected features using criteria like Gini impurity or information gain. For a feature f and threshold t, Gini impurity is given in:

$$\text{Gini(t)} = 1 - \sum(\text{p}_i)^2 \tag{8}$$

where p_i is the proportion of instances of class i in the subset after the split. Information gain is calculated as:

$$\text{Gain(f)} = \text{Entropy(parent)} - \Sigma(|\text{subset}|/|\text{parent}|) * \text{Entropy(subset)} \tag{9}$$

where Entropy(parent) is the entropy of the parent node, |subset| is the number of instances in the subset, and Entropy (subset) is the entropy of the subset. During classification, a new instance is traversed from the root to a leaf node based on feature tests, and the associated class label is assigned as the prediction.

Random Forest Classifier

Random Forest, an ensemble method, combines multiple decision trees to enhance predictive performance. Each tree is built using random subsets of training data and features, mitigating overfitting and enhancing generalization. The classifier integrates predictions from these trees, improving accuracy and generalization. Mathematically, considering dataset X with 'm' instances and 'n' features, the Random Forest construction involves bootstrapping training data and selecting random feature subsets for decision tree splits. The resulting diverse ensemble of trees forms the basis of the classifier, ensuring robustness and predictive power. The classifier h(x) is the mode of individual tree predictions h_1(x), h_2(x), …, h_n(x), determining the final classification through a majority vote in binary or multiclass scenarios.

Gradient Boosting Classifier

The Gradient Boosting Classifier is an ensemble learning method that sequentially builds a strong classifier by combining multiple weak classifiers, typically decision trees. It minimizes a loss function L(y, F(x)) by adjusting the ensemble prediction F(x) in each iteration to fit the negative gradient of the loss ($-\partial L/\partial F(x)$). At each iteration, a new weak classifier h(x) is trained to predict the negative gradient, which helps correct the errors made by the current ensemble. The weak classifiers are added to the ensemble, and their predictions are combined using a weighted sum to make the final prediction. The learning rate (α) controls the contribution of each weak classifier, preventing overfitting and allowing for a gradual learning process. The instance weights are updated iteratively to emphasize misclassified instances, directing the learning towards challenging examples.

K Nearest Neighbours

K Nearest Neighbors (KNN) is a non-parametric algorithm for classification and regression. Given feature matrix X with 'm' instances and 'n' features, KNN classifies a new instance 'x' by finding the K nearest neighbors in X based on distance measures like Euclidean distance. For classification, the majority class among the K neighbors is assigned as the predicted class. For regression, the average of the target values of the K neighbors is used as the prediction.

Mathematically, the Euclidean distance between instances 'x' and '$x'_{\_i}$ in X is:

$$distance(x, x_{\_i}) = \sqrt{\sum_{i=1}^{n} (x - x_i)^2} \tag{10}$$

KNN doesn't involve explicit training; it uses the entire dataset for predictions. The choice of K affects the bias-variance trade-off: smaller K leads to low bias and high variance, while larger K leads to high bias and low variance.

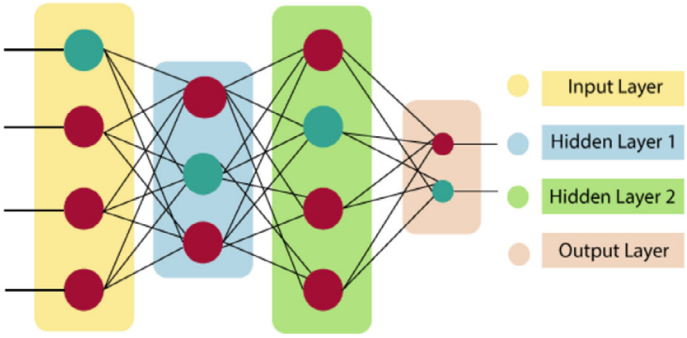


Fig. 3 Diagrammatic representation for an artificial neural network

Artificial Neural Network

An Artificial Neural Network (ANN) comprises interconnected artificial neurons that process inputs using weighted sums and activation functions. The input data consists of a dataset with 'm' instances and 'n' features represented as an m x n matrix X, where each row x_i corresponds to an instance and each column j of x_i represents a feature value x_{ij}. Additionally, let Y be the categorical label matrix, where y_i is the label for instance x_i. As shown in Fig. 3, ANN structure includes an input layer with n neurons, one or more hidden layers with h neurons each, and an output layer with c neurons (for c classes in the classification problem). Each neuron j in a layer is associated with a weight vector w_j and a bias term b_j.

During forward propagation for a single hidden layer, the weighted sum 'z_j' and the output 'a_j' of a neuron j are calculated as: $z_j = \sum(x_{ij} * w_j) + b_j$ for $i = 1$ to m $a_j = f(z_j)$ where f(z) is the activation function applied element-wise to the weighted sum, introducing non-linearity. The activation function is typically chosen based on the problem. For example, the sigmoid function is given by: $f(z) = 1/(1 + \exp(-z))$ and the softmax function used in the output layer for multi-class classification is defined as: $\text{softmax}(z_j) = \exp(z_j)/\sum(\exp(z_i))$ for $i = 1$ to c. During training, the ANN aims to minimize a loss function L, which quantifies the difference between the predicted outputs and the true labels. Commonly used loss functions include categorical cross-entropy: $L = -\sum(y_i * \log(a_i))$ for $i = 1$ to c where a_i is the predicted probability of class i. The weights and biases are adjusted using optimization algorithms like backpropagation, where gradients of the loss function with respect to the weights and biases are computed. The weights are updated using gradient descent or variants like Adam optimization. Once trained, the ANN can classify new instances by passing their input vectors through the network, obtaining class probabilities or labels from the output layer's neurons.

Grid Search and Cross Validation

Initially, each classification algorithm was applied to the training data using default hyperparameter values to establish a baseline. Next, a Grid Search was performed

over each algorithm, exploring different hyperparameter combinations to find the ones that yielded the highest prediction accuracy. During the grid search process, a grid of hyperparameter combinations was specified, and the algorithm was trained and evaluated on each combination. The final stage involved a tenfold cross-validation with a ninety-to-ten ratio, where the dataset was divided into 10 subsets, and classifiers were trained and evaluated on different subsets iteratively. The final prediction accuracy was obtained by averaging the accuracy across the ten folds. This research utilized an i7 desktop computer with an Intel Core i7 processor running at 3.6 GHz and 16 GB of DDR4 RAM.

4 Results

Figure 4 illustrates the LC map for the study region, classified into Built-up, Vegetation, Water body, and Bareland classes. Table 2 illustrate strong agreement between ground features and the classifications made by the Maximum Likelihood Classifier (MLC), exceeding the minimum threshold of 0.80. Specifically, the Kappa values were 0.96 for Water Body, 0.95 for Built-Up, 0.86 for Vegetation, and 0.87 for Bareland, resulting in a total Kappa value of 0.91, indicating consistent agreement across all land cover classes. Additionally, Table 2 presents the area coverage in square kilometers for each LC class. Figure 5 illustrates the NDBI, NDBaI, NDWI, and NDVI maps developed using Eqs. (4–7).

Logistic Regression

The Logistic Regression algorithm executed almost instantaneously in just 0.02 s, producing an initial accuracy of 68.57%. The hyperparameter search range, as shown in Table 3, included ‘C’ values from weak to strong regularization (0.001 to 100), ‘penalty’ options of ‘l1’ and ‘l2’, and ‘solver’ options of ‘lbfgs’, ‘liblinear’, and ‘saga’. The best hyperparameters found were ‘C’: 100, ‘penalty’: l1, and ‘solver’: saga, resulting in a tenfold CV accuracy of approximately 68.51%. The Grid Search process took around 4 min and 48 s to evaluate multiple hyperparameter combinations. Using the optimal hyperparameters, the final classification accuracy improved slightly to 68.76%.

Gaussian Naive Bayes

The initial Gaussian Naive Bayes classification with default settings achieved an accuracy of 64.96% in 0.02 s. The subsequent Grid Search for ‘var_smoothing’ spanned values from 1e-9 to 1e-5 (Table 4). The optimal ‘var_smoothing’ was 1e-09, resulting in a 64.4% CV accuracy in 0.94 s. Despite swift hyperparameter tuning, the optimized settings did not enhance accuracy, maintaining it at 64.96%.

Support Vector Classification

The initial run of the SVC Algorithm took 17.21 s, achieving 69.69% accuracy. After Grid Search optimization with hyperparameters (‘C’: 10, ‘kernel’: ‘rbf’, ‘gamma’:

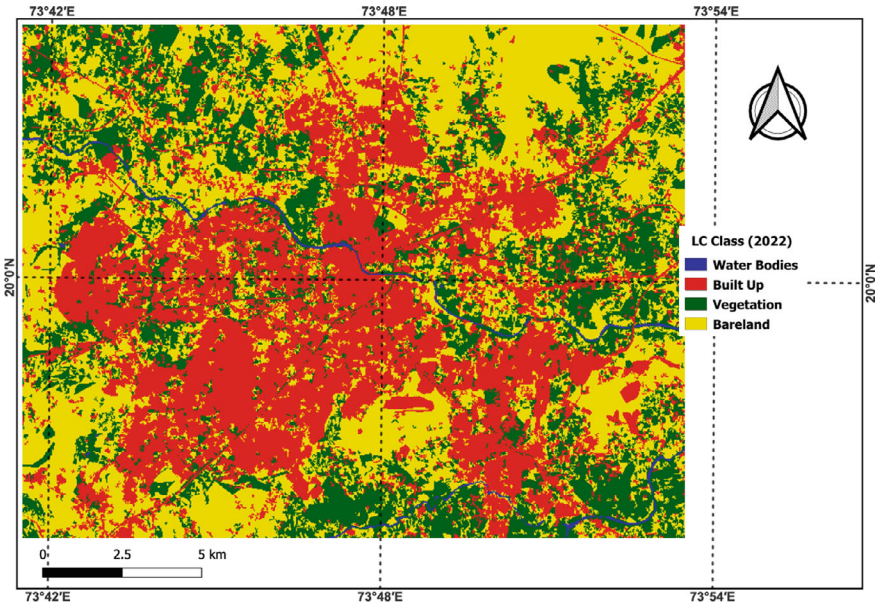


Fig. 4 Land cover map for the Nashik region

Table 2 The computed Kappa statistic and area coverages for the four LC classes

Land cover class	Water body	Built-Up	Vegetation	Bareland	Total
Kappa accuracy	0.96	0.95	0.86	0.87	0.91
Area Coverage (sq km)	2.1	115.5	85.1	133.8	336.52

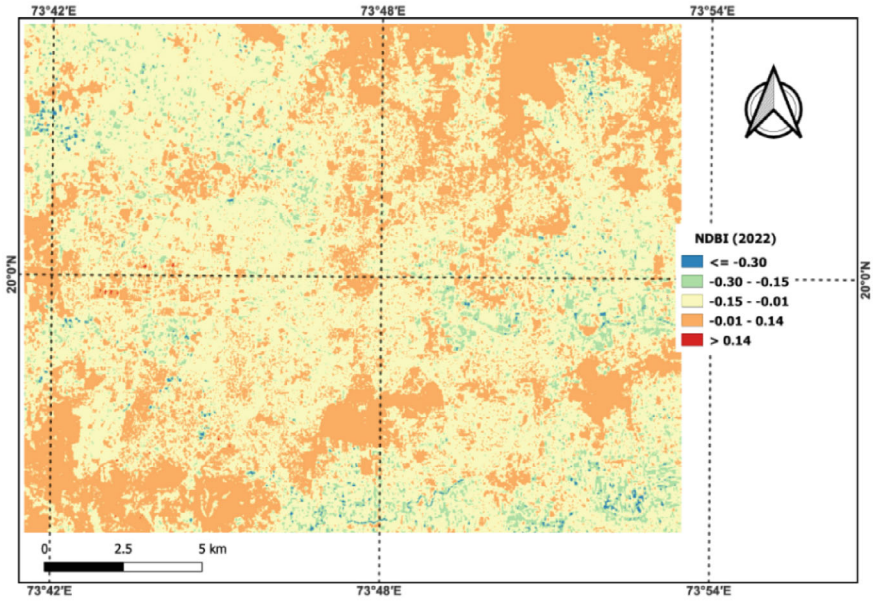
'scale'), the tenfold CV accuracy improved to 69.89% (Table 5). The hyperparameter tuning process lasted 4544.58 s (about 1 h and 27 min). Applying the optimized hyperparameters resulted in a final accuracy of 69.54%.

Decision Tree Classification

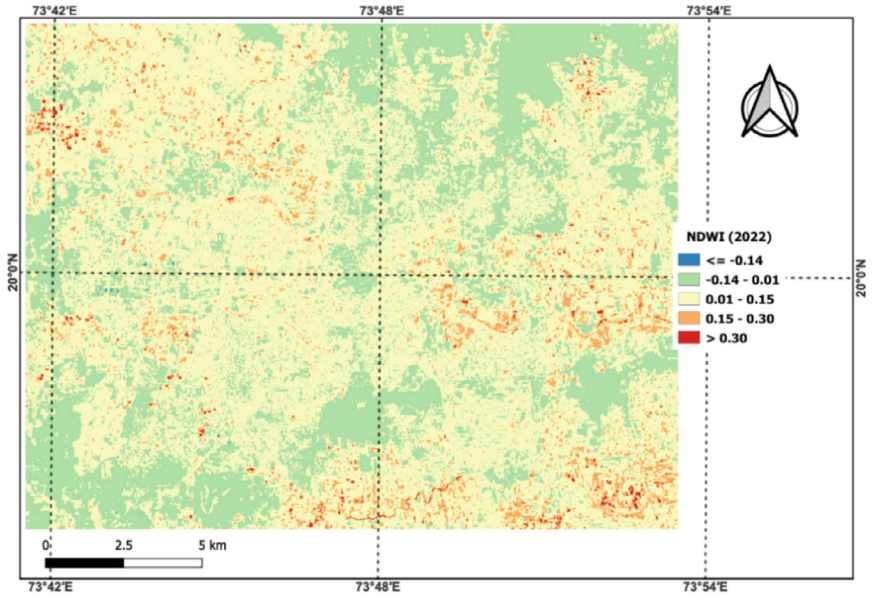
The initial execution of the Decision Tree Classification Algorithm took 0.04 s, resulting in an accuracy of 59.34%. The initial decision tree had 15,441 nodes and 7,721 leaves.

Following Grid Search optimization, the optimal hyperparameters listed in Table 6 were identified as 'criterion': 'entropy', 'max_depth': 10, 'min_samples_split': 10, and 'min_samples_leaf': 4, resulting in a cross-validation accuracy of approximately 68.6%. The hyperparameter tuning process took approximately 218.49 s (about 3 min and 38 s). Figure 6 presents tree diagrams, both complete and zoomed-in, configured with the best hyperparameter settings.

With the optimal hyperparameter settings, the final classification results demonstrated an enhanced accuracy of approximately 68.58%. Additionally, the optimized

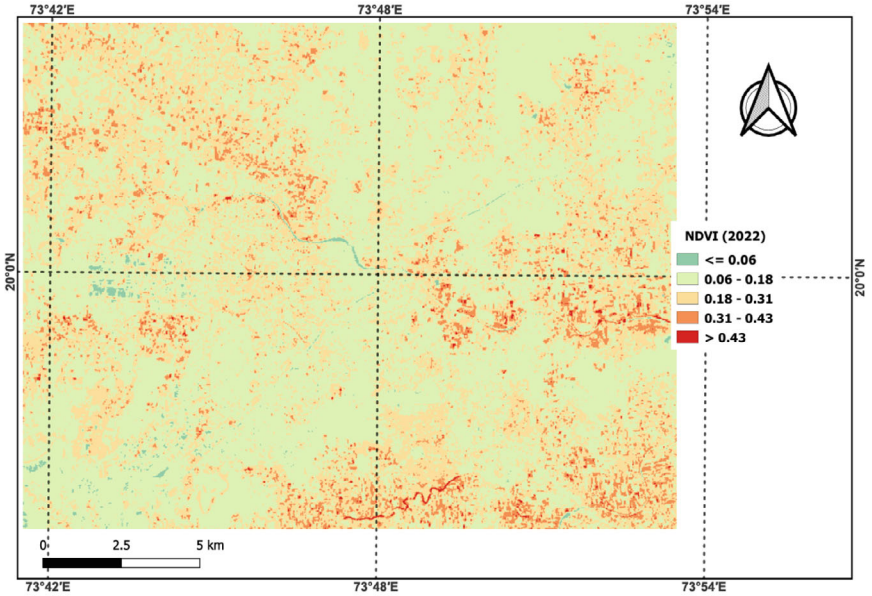


(a)

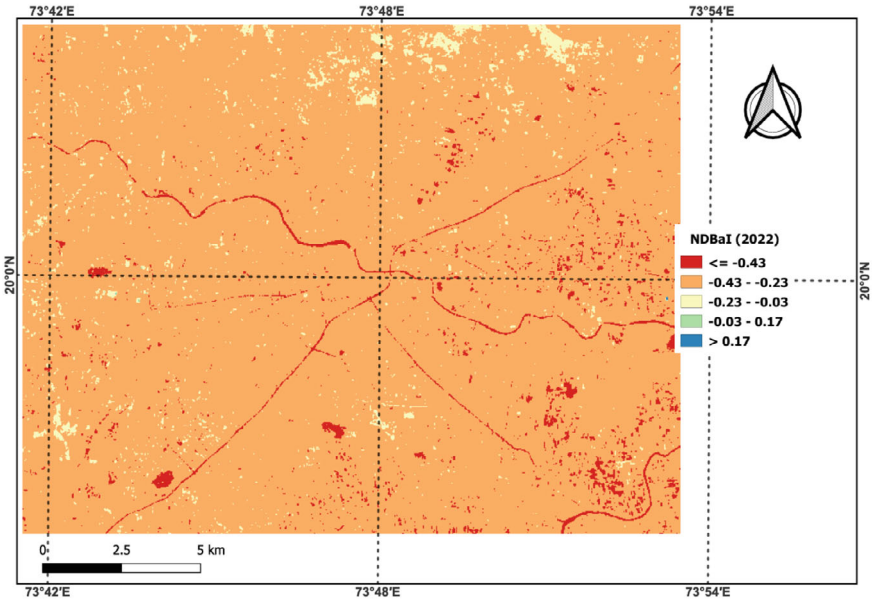


(b)

Fig. 5 **a** NDBI **b** NDBaI **c** NDVI **d** NDWI Maps for the Nashik region



(c)



(d)

Fig. 5 (continued)

Table 3 Grid Search and cross validation results for the Logistic Regression Classifier

Hyperparameter	Options	Optimum values	CV accuracy
'C'	[0.001, 0.01, 0.1, 1, 10, 100]	100	68.51%
'penalty'	['l1', 'l2'],	l1	
'solver'	['lbfgs', 'liblinear', 'saga']	Saga	

Table 4 Grid Search and cross validation results for the Gaussian Naive Bayes Classifier

Hyperparameter	Options	Optimum values	CV Accuracy
'var_smoothing'	[1e-9, 1e-8, 1e-7, 1e-6, 1e-5]	1e-09	0.644

Table 5 Grid Search and cross validation results for the Support Vector Classification Algorithm

Hyperparameter	Options	Optimum values	CV accuracy
'C'	[0.1, 1, 10]	10	0.698
'kernel'	['linear', 'rbf']	scale	
'gamma'	['scale', 'auto']	rbf	

Table 6 Grid Search and cross validation results for Decision Tree Classifier

Hyperparameter	Options	Optimum values	CV accuracy
'criterion'	['gini', 'entropy']	entropy	0.686
'max_depth'	[None, 5, 10, 15],	10	
'min_samples_split'	[2, 5, 10]	10	
'min_samples_leaf'	[1, 2, 4]	4	

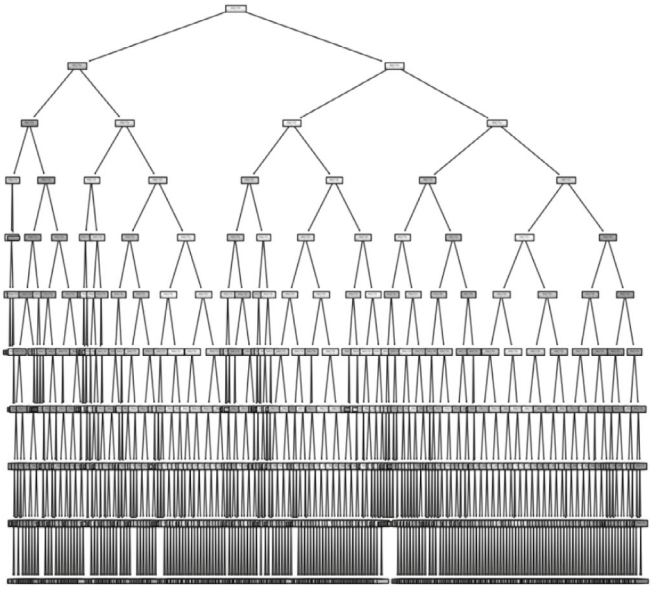
decision tree exhibited a reduced number of nodes (1039) and leaves (520) compared to the initial tree, indicating a more streamlined and efficient model.

Random Forest Classification

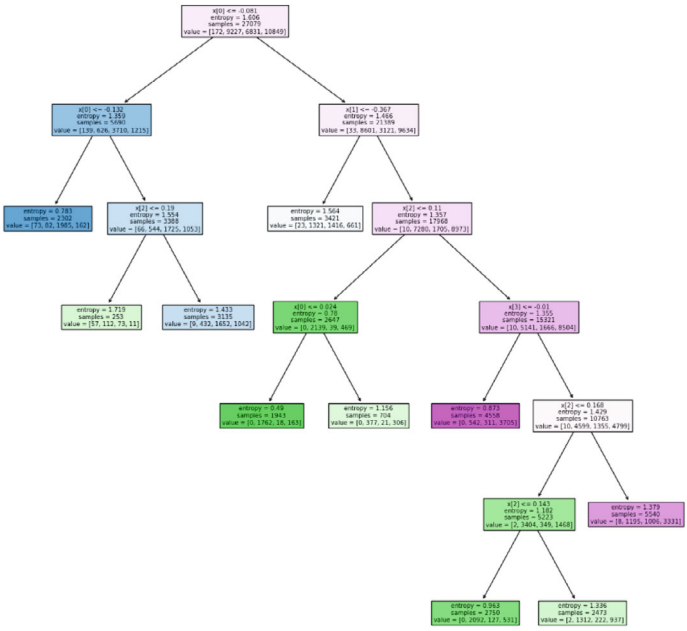
The Random Forest Classification (RFC) initially achieved an accuracy of 67.85% with default settings, executing instantly within 0.25 s. After hyperparameter tuning, as shown in Table 7, the best settings were 'n_estimators': 100, 'max_depth': 10, 'min_samples_split': 5, and 'min_samples_leaf': 1, improving the overall accuracy to 70.26% on the test set (Table 8). The hyperparameter tuning process took 5.58 h. With these optimized hyperparameters, the Random Forest Algorithm achieved a 70.26% overall accuracy for the dataset.

Gradient Boosting Classifier

The initial run of the Gradient Boosting Classifier completed in 0.07 s, achieving a 69.86% accuracy. Table 8 shows the Grid Search results, identifying 'n_estimators': 100, 'learning_rate': 0.1, and 'max_depth': 3 as the best hyperparameters, resulting



(a)



(b)

Fig. 6 **a** Full scale decision tree structure **b** Zoomed-in sectional view of the decision tree using the best hyperparameter structure

Table 7 Grid search and cross validation results for random forest classifier

Hyperparameter	Options	Optimum Values	CV Accuracy
'n_estimators'	[50, 100, 200]	100	0.7026
'max_depth'	[None, 5, 10, 15]	10	
'min_samples_split'	[2, 5, 10]	5	
'min_samples_leaf'	[1, 2, 4]	1	

Table 8 Grid Search hyperparameters and cross validation results for the Gradient Boosting Classifier

Hyperparameter	Options	Optimum Values	CV Accuracy
'n_estimators'	[50, 100, 200]	100	0.704
'learning_rate'	[0.01, 0.1, 0.2]	0.1	
'max_depth'	[3, 5, 7]	3	

in a 70.00% cross-validation accuracy. Despite a lengthy 7.7-h tuning process, there was no improvement over the initial results. The final model, with the best hyperparameters, reached an overall accuracy of 69.96%.

K-Nearest Neighbours Classifier

The initial run of the KNearest Neighbours (KNN) algorithm completed in 0.20 s, achieving an accuracy of 65.74%. The hyperparameter tuning process, detailed in Table 9, lasted 3.82 min, optimizing 'n_neighbors' to 9, and setting 'weights' to 'uniform,' 'algorithm' to 'auto,' and 'metric' to 'euclidean.' This tuning improved the cross-validation accuracy to 67.79%, enhancing the overall accuracy from 65.74% to 67.70%.

Artificial Neural Network

The Artificial Neural Network (ANN) was constructed using the Python Keras library. This Sequential model comprised an input layer with 20 ReLU neurons, a hidden layer with 20 neurons, and a 20% dropout layer. The output layer with 5 sigmoid neurons handles multi-label classification. It was compiled using Adam (lr = 0.01), categorical cross-entropy loss, and the accuracy metric. Figure 7 illustrates the training

Table 9 Grid Search hyperparameters and cross validation results for the KNearest Neighbours Classifier

Hyperparameter	Options	Optimum Values	CV Accuracy
'n_neighbors'	[3, 5, 7, 9]	9	0.677
'weights'	['uniform', 'distance']	Uniform	
'algorithm'	['auto', 'ball_tree', 'kd_tree', 'brute']	Auto	
'metric'	['euclidean', 'manhattan', 'minkowski']	Euclidean	

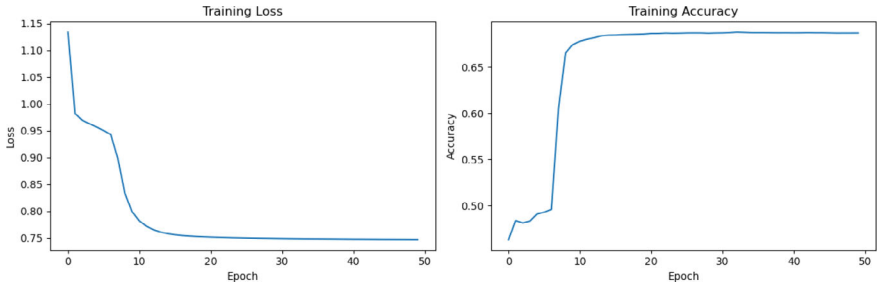


Fig. 7 Training loss and accuracy gains for the ANN model

Table 10 Grid Search hyperparameters and cross validation results for the artificial neural network algorithm

Hyperparameter	Options	Optimum values	CV accuracy
'epochs'	[100, 200, 300]	100	0.7052
'optimizer'	['adam', 'SGD', 'RMSprop']	adam	

accuracy and loss progression over 100 epochs for the ANN model. Table 10 provides the initial classification results for the ANN Classifier Algorithm, which executed in nearly 160 s. The base classification accuracy is 71.23% for the ANN algorithm.

Table 10 displays the ANN hyperparameter options and their optimal values from the Grid Search. 'Epochs' was fine-tuned to 100, and 'optimizer' was set to 'adam,' resulting in a CV accuracy of 71.7%. The exhaustive hyperparameter tuning process took 54.78 s. With these optimal hyperparameters, the ANN prediction accuracy reached 71.23%.

5 Conclusions

The use of robust machine learning techniques enables the establishment of relationships between land use and landscape indicators, such as the Normalized Difference Built-up Index (NDBI), Normalized Difference Bareness Index (NDBaI), Normalized Difference Vegetation Index (NDVI), and Normalized Difference Water Index (NDWI). These techniques facilitate automated Land Use and Land Cover (LULC) mapping in unexplored regions. In this study, LANDSAT 8 satellite imagery for Nashik, India, was analyzed using the Maximum Likelihood Classifier (MLC) algorithm. Landscape indices maps, including NDBI, NDBaI, NDVI, and NDWI, were generated by analyzing reflectance differences in specific frequency bands from the multi-spectral LANDSAT dataset. The datasets were utilized as both training and testing data for eight widely employed classification algorithms: Logistic Regression, Gaussian Naive Bayes, Support Vector Regression, Decision Tree, Random

Forest, Gradient Boosting, K Nearest Neighbor, and Artificial Neural Network. Model precision was assessed for accuracy, hyperparameters were optimized through Grid Search, and predictive efficacy was evaluated using tenfold cross-validation.

The Artificial Neural Network emerged as the most accurate classification algorithm delivering 71.23% accuracy in just 2.67 min, matching the highest accuracy achieved after 54.78 min of hyperparameter tuning. The Support Vector Classifier, starting at 69.69% accuracy, improved to 70.00% post-tuning after 2 h and 49 min. Similarly, the Random Forest algorithm, initially at 67.85% accuracy, notably increased to an impressive 70.26% after a 5.58-h tuning period. Logistic Regression also proved effective, with an initial 68.57% accuracy improving to 68.76% after only 48 s of model tuning. The Gaussian Naive Bayes algorithm achieved an accuracy of 64.96% with default settings, maintaining this accuracy even after hyperparameter tuning. The Support Vector Classification algorithm initially achieved 69.69% accuracy, improving to 69.89% post-tuning after 2 h and 49 min. The Decision Tree algorithm achieved an accuracy of 59.34%, improving to approximately 68.58% after tuning for about 3 min and 38 s. The Gradient Boosting Classifier reached an accuracy of 69.86%, improving slightly to 69.96% post-tuning after 7.7 h. Lastly, the KNearest Neighbours Classifier improved from 65.74% to 67.70% after tuning for 3.82 min.

The study's methodological approach advances remote sensing in urban studies by automating LULC mapping in unexplored regions with robust ML. It emphasizes deriving landscape indices directly from LANDSAT data, eliminating subjectivity. Overall, Logistic Regression, SVM, and Random Forest are identified as effective classifiers for automated land cover mapping in Nashik, India. Gaussian Naive Bayes maintained a lower accuracy compared to other algorithms. SVM showed a moderate improvement, while Decision Tree and Gradient Boosting exhibited similar performance levels post-tuning. KNearest Neighbours also showed a modest improvement in accuracy. One potential limitation of this study was the use of a 30 × 30 m mid-resolution LANDSAT spatial dataset, which, due to its relatively coarse spatial resolution, may not effectively capture fine-scale land cover details. To address this limitation, future studies could consider using higher-resolution datasets like Sentinel-2, which offers 10 × 10 m spatial resolution, or even commercial satellites like WorldView-3, which provide 1 × 1 m data. These higher-resolution datasets can capture finer details on the ground, potentially improving the accuracy and reliability of land cover classification, particularly in areas with complex landscapes or small land cover features.

References

1. Aburas, M.M., Abdullah, S.H., Ramli, M.F., Ash'aari, Z.H.: Measuring land cover change in seremban, Malaysia using NDVI index. Procedia Environ. Sci. **30**, 238–243 (2015).https://doi.org/10.1016/j.proenv.2015.10.043

2. Aburas, M.M., Ahamad, M.S.S., Omar, N.Q.: Spatio-temporal simulation and prediction of land-use change using conventional and machine learning models: a review. Environ. Monitoring Assess. **191**(4) (2019). https://doi.org/10.1007/s10661-019-7330-6
3. AlDousari, A.E., Kafy, A.A., Saha, M., Fattah, M.A., Almulhim, A.I., Faisal, A.A., Al Rakib, A., Jahir, D.M.A., Rahaman, Z.A., Bakshi, A., Shahrier, M., Rahman, M.M.: Modelling the impacts of land use/land cover changing pattern on urban thermal characteristics in Kuwait. Sustain. Cities Soc. **86**, 104107 (2022). https://doi.org/10.1016/j.scs.2022.104107
4. Ali Shah, S., Kiran, M., Nazir, A., Ashrafani, S.H.: Exploring Ndvi and Ndbi relationship using LANDSAT 8 Oli/Tirs in Khangarh Taluka, Ghotki. Malaysian J. Geosci. **6**(1), 8–11 (2022). https://doi.org/10.26480/mjg.01.2022.08.11
5. Alqurashi, A.F., Kumar, L.: Land use and land cover change detection in the Saudi Arabian Desert cities of Makkah and Al-Taif using satellite data. Adv. Remote Sens. **03**(03), 106–119 (2014). https://doi.org/10.4236/ars.2014.33009
6. Ao, Y., Xu, B.: Information extraction of urban expansion based on remote sensing. ISWREP 2011—Proceedings of 2011 International Symposium on Water Resource and Environmental Protection, 4(40971217), 2683–2686 (2011). https://doi.org/10.1109/ISWREP.2011.5893431
7. Arulbalaji, P.: Analysis of land use/land cover changes using geospatial techniques in Salem district, Tamil Nadu, South India. SN Appl. Sci. **1**(5) (2019). https://doi.org/10.1007/s42452-019-0485-5
8. Aswatha, S.M., Mukherjee, J., Biswas, P.K., Aikat, S.: Toward automated land cover classification in LANDSAT images using spectral slopes at different bands. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. **10**(3), 1096–1104 (2017). https://doi.org/10.1109/JSTARS.2016.2602390
9. Bouzekri, S., Lasbet, A.A., Lachehab, A.: A new spectral index for extraction of built-up area using LANDSAT-8 data. J. Indian Soc. Remote Sens. **43**(4), 867–873 (2015). https://doi.org/10.1007/s12524-015-0460-6
10. Chaturvedi, S., Shukla, K., Rajasekar, E., Bhatt, N.: A spatio-temporal assessment and prediction of Ahmedabad's urban growth between 1990–2030. J. Geog. Sci. **32**(9), 1791–1812 (2022). https://doi.org/10.1007/s11442-022-2023-4
11. Chen, X., Yang, D., Chen, J., Cao, X.: An improved automated land cover updating approach by integrating with downscaled NDVI time series data. Remote Sens. Lett. **6**(1), 29–38 (2015). https://doi.org/10.1080/2150704X.2014.998793
12. Chopade, M.R., Mahajan, S., Chaube, N.: Assessment of land use, land cover change in the mangrove forest of Ghogha area, Gulf of Khambhat, Gujarat. Expert Syst. Appl. **212**(September 2022), 118839 (2023). https://doi.org/10.1016/j.eswa.2022.118839
13. Choudhury, D., Das, K., Das, A.: Assessment of land use land cover changes and its impact on variations of land surface temperature in Asansol-Durgapur development region. Egypt. J. Remote Sens. Space Sci. **22**(2), 203–218 (2019). https://doi.org/10.1016/j.ejrs.2018.05.004
14. Chowdhury, S., Peddle, D.R., Wulder, M.A., Heckbert, S., Shipman, T.C., Chao, D.K.: Estimation of land-use/land-cover changes associated with energy footprints and other disturbance agents in the Upper Peace Region of Alberta Canada from 1985 to 2015 using LANDSAT data. Int. J. Appl. Earth Observ. Geoinf. **94**(August 2020), 102224 (2021). https://doi.org/10.1016/j.jag.2020.102224
15. Das, S., Angadi, D.P.: Land use-land cover (LULC) transformation and its relation with land surface temperature changes: a case study of Barrackpore Subdivision, West Bengal, India. Remote Sens. Appl. Soc. Environ. **19**, 100322 (2020). https://doi.org/10.1016/j.rsase.2020.100322
16. Fahad, S., Li, W., Lashari, A.H., Islam, A., Khattak, L.H., Rasool, U.: Evaluation of land use and land cover Spatio-temporal change during rapid Urban sprawl from Lahore. Pakistan. Urban Climate **39**, 100931 (2021). https://doi.org/10.1016/j.uclim.2021.100931
17. Faruque, M.J., Hasan, M.Y., Islam, K.Z., Young, B., Ahmed, M.T., Monir, M.U., Shovon, S.M., Kakon, J.F., Kundu, P.: Monitoring of land use and land cover changes by using remote sensing and GIS techniques at human-induced mangrove forests areas in Bangladesh. Remote Sens. Appl. Soc. Environ. **25**, 100699 (2022). https://doi.org/10.1016/j.rsase.2022.100699

18. Guha, S., Govil, H., Dey, A., Gill, N.: Analytical study of land surface temperature with NDVI and NDBI using LANDSAT 8 OLI and TIRS data in Florence and Naples city Italy. Europ. J. Remote Sens. **51**(1), 667–678 (2018). https://doi.org/10.1080/22797254.2018.1474494
19. Halder, B., Bandyopadhyay, J., Banik, P.: Evaluation of the climate change impact on urban heat island based on land surface temperature and geospatial indicators. Int. J. Environ. Res. **15**(5), 819–835 (2021). https://doi.org/10.1007/s41742-021-00356-8
20. Han, H., Yang, C., Song, J.: Scenario simulation and the prediction of land use and land cover change in Beijing. China. Sustainability (Switzerland) **7**(4), 4260–4279 (2015). https://doi.org/10.3390/su7044260
21. Hu, Y., Dong, Y., Batunacun: An automatic approach for land-change detection and land updates based on integrated NDVI timing analysis and the CVAPS method with GEE support. ISPRS J. Photogr. Remote Sens. 146, 347–359 (2018). https://doi.org/10.1016/j.isprsjprs.2018.10.008
22. Hussain, S., Mubeen, M., Ahmad, A., Akram, W., Hammad, H.M., Ali, M., Masood, N., Amin, A., Farid, H.U., Sultana, S.R., Fahad, S., Wang, D., Nasim, W.: Using GIS tools to detect the land use/land cover changes during forty years in Lodhran District of Pakistan. Environ. Sci. Pollut. Res. **27**(32), 39676–39692 (2020). https://doi.org/10.1007/s11356-019-06072-3
23. Hussain, S., Mubeen, M., Karuppannan, S.: Land use and land cover (LULC) change analysis using TM, ETM+ and OLI LANDSAT images in district of Okara, Punjab, Pakistan. Phys. Chem. Earth **126**, 103117 (2022). https://doi.org/10.1016/j.pce.2022.103117
24. Jhonnerie, R., Siregar, V.P., Nababan, B., Prasetyo, L.B., Wouthuyzen, S.: Random forest classification for mangrove land cover mapping using LANDSAT 5 TM and Alos Palsar Imageries. Procedia Environ. Sci. **24**, 215–221 (2015). https://doi.org/10.1016/j.proenv.2015.03.028
25. Jia, K., Liang, S., Wei, X., Yao, Y., Su, Y., Jiang, B., Wang, X.: Land cover classification of LANDSAT data with phenological features extracted from time series MODIS NDVI data. Remote Sens. **6**(11), 11518–11532 (2014). https://doi.org/10.3390/rs61111518
26. Kafy, A.A., Saha, M., Faisal, A.A., Rahaman, Z.A., Rahman, M.T., Liu, D., Fattah, M.A., Al Rakib, A., AlDousari, A.E., Rahaman, S.N., Hasan, M.Z., Ahasan, M.A.K.: Predicting the impacts of land use/land cover changes on seasonal urban thermal characteristics using machine learning algorithms. Build. Environ. **217**, 109066 (2022). https://doi.org/10.1016/j.buildenv.2022.109066
27. Khalmurzayeva, G.: Land cover classification using random forest technique and DEM auxiliary data. Int. J. Geoinform. **15**(4), 1–9 (2019)
28. Krishnaveni, K.S., Anilkumar, P.P.: A Fully Automated approach to extract landcover features from LANDSAT imageries. Int. Geosci. Remote Sens. Symp. (IGARSS) **3**, 6669–6672 (2021). https://doi.org/10.1109/IGARSS47720.2021.9554848
29. Kulkarni, A.D., Lowe, B.: Random forest algorithm for land cover classification. Int. J. Recent Innov. Trends Comput. Commun. **4**(3), 58–63 (2016)
30. Kulkarni, K., Vijaya, P.: NDBI based prediction of land use land cover change. J. Indian Soc. Remote Sens. **49**(10), 2523–2537 (2021). https://doi.org/10.1007/s12524-021-01411-9
31. Li, E., Du, P., Samat, A., Xia, J., Che, M.: An automatic approach for urban land-cover classification from LANDSAT-8 OLI data. Int. J. Remote Sens. **36**(24), 5983–6007 (2015). https://doi.org/10.1080/01431161.2015.1109726
32. Liu, X., Hu, G., Chen, Y., Li, X., Xu, X., Li, S., Pei, F., Wang, S.: High-resolution multi-temporal mapping of global urban land using LANDSAT images based on the Google Earth Engine Platform. Remote Sens. Environ. **209**, 227–239 (2018). https://doi.org/10.1016/j.rse.2018.02.055
33. Macarringue, L.S., Bolfe, É.L., Pereira, P.R.M.: Developments in land use and land cover classification techniques in remote sensing: a review. J. Geogr. Inf. Syst. **14**(01), 1–28 (2022). https://doi.org/10.4236/jgis.2022.141001
34. Madasa, A., Orimoloye, I.R., Ololade, O.O.: Application of geospatial indices for mapping land cover/use change detection in a mining area. J. African Earth Sci. **175**(August 2020), 104108 (2021). https://doi.org/10.1016/j.jafrearsci.2021.104108
35. Meshesha, T.W., Tripathi, S.K., Khare, D.: Analyses of land use and land cover change dynamics using GIS and remote sensing during 1984 and 2015 in the Beressa Watershed

Northern Central Highland of Ethiopia. Model. Earth Syst. Environ. **2**(4), 1–12 (2016). https://doi.org/10.1007/s40808-016-0233-4
36. Moumane, A., Al Karkouri, J., Benmansour, A., El Ghazali, F.E., Fico, J., Karmaoui, A., Batchi, M.: Monitoring long-term land use, land cover change, and desertification in the Ternata oasis, Middle Draa Valley, Morocco. Remote Sens. Appl. Soc. Environ. **26**(October 2021), 100745 (2022). https://doi.org/10.1016/j.rsase.2022.100745
37. Mubako, S., Nnko, H.J., Peter, K.H., Msongaleli, B.: Evaluating historical and predicted long-term land use/land-cover change in Dodoma Urban District, Tanzania: 1992–2029. Phys. Chem. Earth **128**, 103205 (2022). https://doi.org/10.1016/j.pce.2022.103205
38. Nurwanda, A., Honjo, T.: The prediction of city expansion and land surface temperature in Bogor City, Indonesia. Sustain. Cities Soc. **52**(December 2018), 101772 (2020). https://doi.org/10.1016/j.scs.2019.101772
39. Parent, J.R., Volin, J.C., Civco, D.L.: A fully-automated approach to land cover mapping with airborne LiDAR and high resolution multispectral imagery in a forested suburban landscape. ISPRS J. Photogramm. Remote. Sens. **104**, 18–29 (2015). https://doi.org/10.1016/j.isprsjprs.2015.02.012
40. Parmar, M., Bhawsar, Z., Kotecha, M., Shukla, A., Rajawat, A.S.: Assessment of land degradation vulnerability using geospatial technique: a case study of Kachchh District of Gujarat, India. J. Indian Soc. Remote Sens. **49**(7), 1661–1675 (2021). https://doi.org/10.1007/s12524-021-01349-y
41. Piyoosh, A.K., Ghosh, S.K.: Semi-automatic mapping of anthropogenic impervious surfaces in an urban/suburban area using LANDSAT 8 satellite data. GIScience Remote Sens. **54**(4), 471–494 (2017). https://doi.org/10.1080/15481603.2017.1282414
42. Rahman, M.T.: Detection of land use/land cover changes and urban sprawl in Al-Khobar, Saudi Arabia: an analysis of multi-temporal remote sensing data. ISPRS Int. J. Geo-Info. **5**(2) (2016). https://doi.org/10.3390/ijgi5020015
43. Roy, S. Sen, Rahman, A., Ahmed, S., Shahfahad, Ahmad, I.A.: Long-term trends of groundwater level variations in response to local level land use land cover changes in Mumbai, India. Groundwater Sustain. Dev. **18**(December 2021) (2022). https://doi.org/10.1016/j.gsd.2022.100797
44. Shukla, A., Jain, K.: Modeling urban growth trajectories and spatiotemporal pattern: a case study of Lucknow City, India. J. Indian Soc. Remote Sens. **47**(1), 139–152 (2019). https://doi.org/10.1007/s12524-018-0880-1
45. Singh, P., Verma, P., Chaudhuri, A.S., Singh, V.K., Rai, P.K.: Evaluating the relationship between Urban Heat Island and temporal change in land use, NDVI and NDBI: a case study of Bhopal city, India. Int. J. Environ. Sci. Technol., 0123456789 (2023).https://doi.org/10.1007/s13762-023-05141-y
46. Szabó, S., Gácsi, Z., Balázs, B.: Specific features of NDVI, NDWI and MNDWI as reflected in land cover categories. Landscape Environ. **10**(3–4), 194–202 (2016). https://doi.org/10.21120/le/10/3-4/13
47. Tariq, A., Yan, J., Mumtaz, F.: Land change modeler and CA-Markov chain analysis for land use land cover change using satellite data of Peshawar, Pakistan. Phys. Chem. Earth **128**, 103286 (2022). https://doi.org/10.1016/j.pce.2022.103286
48. Thakur, S., Maity, D., Mondal, I., Basumatary, G., Ghosh, P.B., Das, P., De, T.K.: Assessment of changes in land use, land cover, and land surface temperature in the mangrove forest of Sundarbans, northeast coast of India. Environ. Dev. Sustain. **23**(2), 1917–1943 (2021). https://doi.org/10.1007/s10668-020-00656-7
49. Varshney, A., Rajesh, E.: A comparative study of built-up index approaches for automated extraction of built-up regions from remote sensing data. J. Indian Soc. Remote Sens. **42**(3), 659–663 (2014). https://doi.org/10.1007/s12524-013-0333-9
50. Xu, H.: Extraction of urban built-up land features from LANDSAT imagery using a thematic-oriented index combination technique. Photogra. Eng. Remote Sens. **73**(12), 1381–1391 (2007). https://doi.org/10.14358/PERS.73.12.1381

51. Zha, Y., Gao, J., Ni, S.: Use of normalized difference built-up index in automatically mapping urban areas from TM imagery. Int. J. Remote Sens. **24**(3), 583–594 (2003). https://doi.org/10.1080/01431160304987
52. Zhao, L., Zhang, P., Ma, X., Pan, Z.: Land cover information extraction based on daily NDVI time series and multiclassifier combination. Math. Problems Eng. (2017).https://doi.org/10.1155/2017/6824051