Technologien für die intelligente Automation
Technologies for Intelligent Automation

Oliver Niggemann
Jürgen Beyerer
Maria Krantz
Christian Kühnert *Editors*

Machine Learning for Cyber-Physical Systems

Selected papers from the International Conference ML4CPS 2023

Springer

Technologien für die intelligente Automation

Technologies for Intelligent Automation

Volume 18

Reihe herausgegeben von

Institut für industrielle Informationstechnik – inIT, Technische Hochschule Ostwestfalen-Lippe, Institut für industrielle Informationstechnik – inIT, Lemgo, Germany

Ziel der Buchreihe ist die Publikation neuer Ansätze in der Automation auf wissenschaftlichem Niveau, Themen, die heute und in Zukunft entscheidend sind, für die deutsche und internationale Industrie und Forschung. Initiativen wie Industrie 4.0, Industrial Internet oder Cyber-physical Systems machen dies deutlich. Die Anwendbarkeit und der industrielle Nutzen als durchgehendes Leitmotiv der Veröffentlichungen stehen dabei im Vordergrund. Durch diese Verankerung in der Praxis wird sowohl die Verständlichkeit als auch die Relevanz der Beiträge für die Industrie und für die angewandte Forschung gesichert. Diese Buchreihe möchte Lesern eine Orientierung für die neuen Technologien und deren Anwendungen geben und so zur erfolgreichen Umsetzung der Initiativen beitragen.

Oliver Niggemann · Jürgen Beyerer ·
Maria Krantz · Christian Kühnert
Editors

Machine Learning for Cyber-Physical Systems

Selected papers from the International Conference ML4CPS 2023

Springer

Editors
Oliver Niggemann
Fakultät für Maschinenbau
Helmut Schmidt University
Hamburg, Germany

Maria Krantz
Fakultät für Maschinenbau
Helmut Schmidt University
Hamburg, Germany

Jürgen Beyerer
Fraunhofer IOSB
Karlsruhe, Germany

Christian Kühnert
Fraunhofer IOSB
Karlsruhe, Germany

ISSN 2522-8579 ISSN 2522-8587 (electronic)
Technologien für die intelligente Automation
ISBN 978-3-031-47061-5 ISBN 978-3-031-47062-2 (eBook)
https://doi.org/10.1007/978-3-031-47062-2

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

Paper in this product is recyclable.

Preface

Artificial intelligence and especially machine learning are becoming more and more commonplace. This is mainly due to higher data availability and more sophisticated tools. It becomes more difficult when these ML methods are applied to cyber-physical systems. Often less data is available, algorithms are less understood and issues like reliability and testability become crucial. These topics were discussed at the 6th ML4CPS—Machine Learning for Cyber-Physical Systems Conference from March 29th to 31st in Hamburg. Experts from industry and research discussed the state of the art and listened to presentations on new developments. The conference was jointly organized by the Helmut-Schmidt-University, Fraunhofer IOSB and ARIC.

Oliver Niggemann
Jürgen Beyerer
Maria Krantz
Christian Kühnert

Contents

Causal Structure Learning Using PCMCI+ and Path Constraints from Wavelet-Based Soft Interventions

Josephine Rehak, Alexander Falkenstein and Jürgen Beyerer

Abstract

The discovery of causal relations via interventions has proven to be simple when only one observed variable is affected or unaffected. However, in a multivariate setting, it is likely that more than one variable is affected by an intervention and thus drawing conclusions about the causal relations becomes more difficult as the gained information is ambiguous. To deal with this, we introduce a novel definition of path constraints and to obtain them, we came up with a novel approach for wavelet-based interventions. We demonstrate our approach on a combustion engine simulation, where we injected wavelets of our choice in an actuated variable and tried to rediscover them in the other, observed variables to gain path constraints. Subsequently, we demonstrate how to use these constraints to optimize the results of the established PCMCI+ algorithm.

Schlüsselwörter

Causal structure learning • Soft interventions • Wavelets • Constraint-based structure learning • Pattern matching

J. Rehak (✉) · A. Falkenstein
Karlsruhe Institute of Technology (KIT), Vision and Fusion Laboratory, Karlsruhe, Germany
e-mail: Josephine.Rehak@kit.edu

J. Beyerer
Fraunhofer IOSB, Karlsruhe, Germany
e-mail: juergen.beyerer@iosb.fraunhofer.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien
für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_1

1 Introduction

The domain of causal structure learning deals with finding cause and effect relationships in inspected environments and aims to find the true causal structure graph. These novel machine learning methods can help a machine to independently understand its environment, to use its causal dependencies, and also to assess the consequences of actions that have already occurred or will occur in the future. The discovery of the causal graphs is used for causal-motivated root cause analysis [5] and causal effect estimation in smart manufacturing. For their development, it is common to apply them on well-known environments.

A subdomain of causal structure learning tries to uncover causal relations by performing experiments and gaining structure knowledge by inspecting the interventional effects. But established procedures as the Rubin causal model [9], the basic theory for randomized controlled trials [3], inspect the effect in only **one** affected variable. Currently, gaining structure knowledge from interventions that affected multiple variables is under discovered, since the provided information about the causal graph is highly ambiguous. Each causal relation between the intervened and the affected variable could be direct, or it could be indirect via any other affected variable.

In this work, we investigated how to use such ambiguous information for causal structure learning by defining so called path constraints. In a novel low-intrusive intervention, we injected a minor wavelet onto the intervened variable, tried to rediscover the wavelet to identify the affected variables and thus derived these path constraints. A path constraint contains the sole information that two variables are direct or indirect connected somehow in a specified direction. We demonstrate their usefulness by applying them to the results of the state of the art PCMCI+ causal structure learning algorithm and thereby improve its discovered causal graphs.

In order to exercise these methods, we have chosen a simulation of a running combustion engine as testing environment. The specific chosen combustion engine was validated on a test stand, has a manageable and well-known number of causal relationships. Its simulation allows the injection of wavelets that can propagate in the system naturally.

The paper is structured as follows. In Sect. 2, we will present related work. In Sect. 3, we shed some light on causal graphs and existing approaches for causal structure learning. In Sect. 4, we introduce our fundamentals for the novel interventional technique. In Sect. 5, the wavelet injections are demonstrated on a combustion engine simulation step by step. In Sect. 5, we draw the conclusion.

2 Related Work

Related Work in Soft Interventions

In the domain of soft interventions, the use of probabilistic interventions is common to gain structure information between two investigated variables. Eberhardt et al. [2] investigated how many soft interventions in variable pairs are required to gain knowledge over the true full causal graph. Kühnert et al. [6] exercised the Max min parents and children method to learn the graph skeleton, oriented some edges using conditional independence tests and then applied soft interventions by 'pushing' probability distributions in a certain direction to finalize the discovered graph by inspecting variable pairs.

Related Work in Using Propagation Properties
A related characteristic about the propagation properties of causal relations has been used before by Hoyer et al. [4]. According to their research, the affected variable may be represented as a function of the causing variable and some random and independent additive noise in the actual direction from cause to effect, but not in the opposed direction. By testing for each direction, they were able to direct the relations between the two variables with high confidence.

Related Work in Multivariate Discovery
Some methods have specialized in the discovery of multiple causal relations to uncover the causal graph such as such as PCMCI+ [10] and Multivariate Transfer Entropy [7]. They used only observed data for their discoveries and performed no interventions.

3 Fundamentals

3.1 Causal Graphs

Causal graphs consist of a set of nodes representing variables $\mathcal{V}$ and a set of edges $\mathcal{E}$ representing causal relations. If a directed edge points from $A \in \mathcal{V}$ to $B \in \mathcal{V}$, then variable B is caused by A. A path from A to B is a chain of *arbitrary* edges connecting A and B with a number of edges being equal or greater than one. A *directed* path from A to B is a path with consistently directed edges from A to B. A direct causal relation between variables indicates a path length of one, but an indirect causal relation indicates a path length greater than one.

3.2 Causal Structure Learning

The goal of causal structure learning is to gain knowledge over the true causal graph of the inspected environment. Such knowledge may be gained by applying observational and interventional methods.

Observational Causal Structure Learning Observational causal structure learning methods try to find knowledge over the true causal graph from recorded data gained by observations. One of the most powerful methods has become PCMCI+ [10]. It works in two phases. First, a set of potential parents is estimated in a specified time window. In the second step, a Mutual Conditional Indepencence (MCI) test is applied on the variables from the parent set and from the inspected time window. Opposed to prior algorithms, PCMCI+ does not assume causal edges to be oriented either in the one or the other direction. Thus, it allows finding bidirected and contemporaneous causal relations as they are often present in engineering applications.

Interventional Causal Structure Learning Using interventions for causal structure learning is one of the oldest and most popular approaches in science. Still in some application scenarios their use may be costly, unethical or simply not feasible. It is assumed that a variable is a cause of another variable B, if an intervention on A also affects the associated variable B [12]. [2] distributed the existing approaches in two major categories called structural interventions and soft interventions. Structural interventions cut off all causal influences to the variable under intervention and fully determine its value (e.g. treatment drug or no drug). Soft interventions (also called parametric interventions) only undertake minor changes. Common is the intervention on the probability distribution of a variable [2, 6]. They do not disturb the original causal structure, therefore influences of other variables on the intervened variable are not ruled out.

4 Wavelet-Based Soft Interventions

The new intervention method does not perturb the original causal relation and can therefore be considered according to Sect. 3 as soft intervention. The particularity of these interventions lies in the fact that a wavelet is added to the intervened variable and is tried to be rediscovered to gain causal information.

When injecting a wavelet into a variable A, the injected wavelet is added to the timeseries of the injected variable. We assume the wavelet to spread in direction of the causal relations in the graph. If we find the wavelet in only one variable, we assume a direct causal relation to be present. In case of a discovery in several other variables, including B, we may not. Instead, we gain knowledge about an existing path between the variables with $|C(A, B)| \geq 1$, since the wavelet must have traveled in some way from A to B. This specific path information are the so called path constraints.

As soft interventions allow other causal influences on the intervened variable, the wavelet was required to be distinguishable from the other influences. Thus, we decided on the use of uniquely shaped wavelets to increase the chance of their successful reidentification. Note, that in general, we do not consider information about variables in which the injected wavelet could **not** be found, as the wavelet may be lost due to loss in amplitude, deformation or other reasons.

For wavelet recovery, we normalized the measured values. Otherwise, the different scales of the variables would make a comparison difficult. Then we applied on each measured variable the fast pattern matching algorithm called Mueen's ultra-fast Algorithm for Similarity Search (MASS) [13]. It gradually matches a desired pattern to a subsequence of the inspected timeseries and calculates the z-normalized Euclidean distance. The aggregation of these distances results in an overall distance profile. If its minimal distance is below a chosen threshold, we assume the position to be our wavelet. Otherwise, we assume the wavelet to be absent in the observed variable and thus we gain no path constraint.

5 Applying Wavelet Injections

In this section, we demonstrate how we applied the wavelet injections on a combustion engine dataset. As experimental setup, we first applied PCMCI+ on the simulation timeseries without any wavelet injections present. Then, we added three different wavelets to a root variable and tried to rediscover them in the other variables to gain path constraints. Later on, we compared the discovered graphs found by PCMCI+ and PCMCI+ combined with path constraints against the actual causal graph using the Receiver Operating Curve Area under Curve (ROC AUC). For this purpose, the true causal graph was created in advance using expert knowledge.

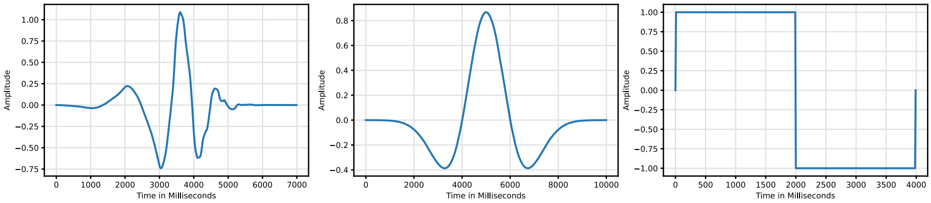
Simulation Setup

As a testing environment, we used a running combustion engine simulation [1, 8, 11]. For evaluation, we constructed the true causal graph as is shown in Fig. 2. Here, we give a brief explanation of the causal relations: The *angle* of the throttle plate influences how much *air intake* in the motor cylinder is possible. The *air intake* over time adds up to the *aircharge* in the cylinder before combustion. After combustion, depending on the *aircharge*, increases the *torque* of the engine and the overall *engine speed*. The increase in *engine speed* also depends on the *load* carried by the engine.

We injected the wavelets shown in Fig. 1 by actuating only the throttle *angle* and inspected all the other measured timeseries for traces of the injected wavelet. According to the true causal graph in Fig. 2, the wavelets should be found in the *air intake*, *aircharge*, *torque* and *engine speed* variable, as they directly and indirectly depend on the *angle* variable. Only the *load* variable should be free of any wavelet, as it is independent of the *angle* variable.

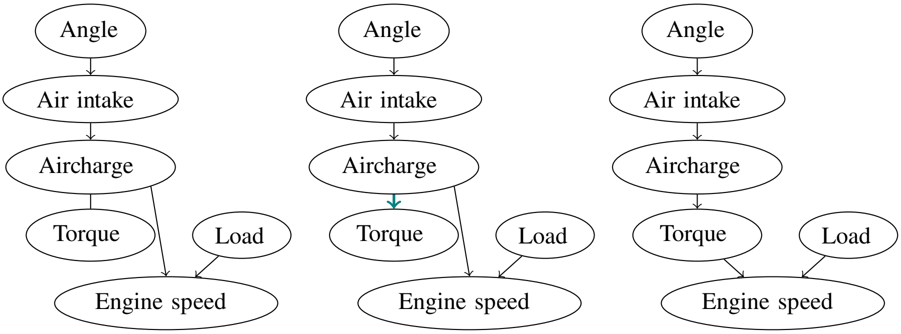
Step 1: Performing PCMCI+

First, the PCMCI+ described in Sect. 3.2 was applied on the running combustion engine dataset. We performed six measurements by setting the alpha value to 0.05 and 0.01 for each maximum lag parameter of 5, 10 and 15 data entries. One of these graphs is depicted in Fig. 2a as an example. In average the graphs found by PCMCI+ achieved an average ROC



(a) Daubechie 4 wavelet (b) Mexican hat wavelet (c) Haar wavelet

Fig. 1 The wavelets that were injected into the *angle* variable



(a) PCMCI+ (b) PCMCI+ with injection (c) True causal graph

Fig. 2 Depicted are the causal graphs **a**) after the application of PCMCI+; **b**) after the application of the path constraints to the PCMCI+ result; and **c**) the actual causal graph constructed by experts

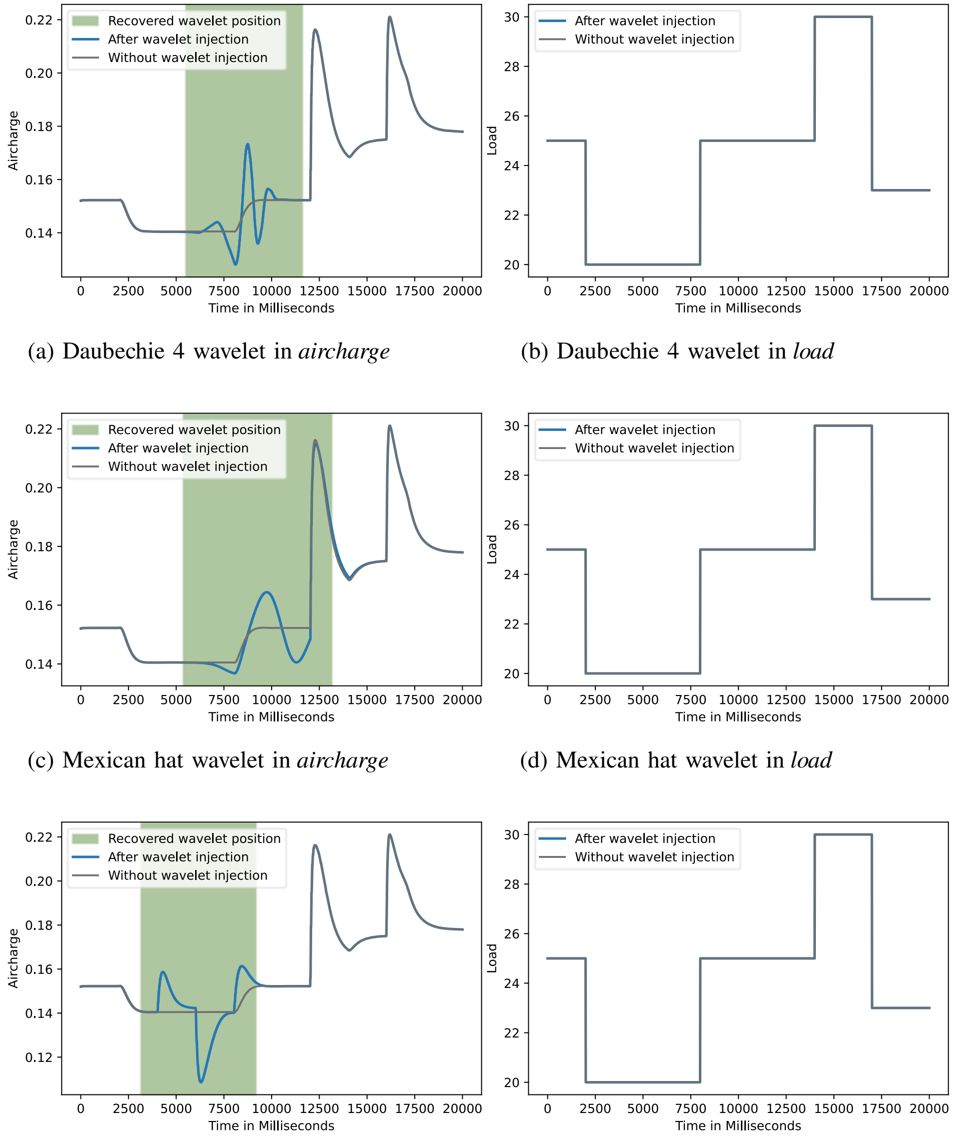
AUC of 73% points in respect to the true causal graph shown in Fig. 2c. As shown, the method was not able to direct several edges.

Step 2: Performing Wavelet Injection and Recovery

We decided to use three very distinct and well-defined wavelets: a Daubechie 4 wavelet, a Mexican Hat wavelet and a Haar wavelet. They are depicted in Fig. 1. We have chosen these wavelets because they contain amplitudes in the positive and negative value range and have a distinct shape.

As an implementation of the pattern matching algorithm, we used the python package stumpy[1]. It found all wavelets in all variables depending on the influenced *angle* variable. Each wavelet was sufficient enough for injection and discovery, as each was found in the actual position in all variables depending on the *angle* variable. Figure 3 presents an excerpt from our results for the *aircharge* and the *load* variable for each of the three wavelets. Plotted are the measured variables with and without variable injection, thus any divergence between the plotted lines must be caused by the wavelet. The colored area is where the wavelet was

[1] https://stumpy.readthedocs.io/



(a) Daubechie 4 wavelet in *aircharge*

(b) Daubechie 4 wavelet in *load*

(c) Mexican hat wavelet in *aircharge*

(d) Mexican hat wavelet in *load*

(e) Haar wavelet in *aircharge*

(f) Haar wavelet in *load*

Fig. 3 The wavelets as they were discovered in the exemplary *aircharge* and *load* variable. The area where the lines diverge indicates the presence of a wavelet and should be highlighted green as a mark for successful recovery. If the lines do not diverge, no wavelet is present and nothing should be discovered.

rediscovered by the pattern matching algorithm. It is colored green, then the wavelet is found in its actual position. This was the case for all wavelets in the *aircharge* variable.
In the *load* variable, no wavelet was found as it should be, since the variable is not dependent on the *angle* variable.

Step 3: Combining PCMCI+ and Path Constraints
From the previous step, we were able to retrieve four path constraints as the wavelet injection in *angle* affected the *air intake*, *aircharge*, *torque*, and *engine speed* variable. To improve the results of PCMCI+, we oriented the unoriented edges in the found causal graph according to the path constraints. We assumed a directed path to be present between the causing and the affected variable for each path constraint. If only one path was present in the PCMCI+ result, we directed it's undirected edges in accordance with the directed path. We applied this procedure on all six PCMCI+ results. An example is shown in Fig. 2b. Due to the path constraints, the ROC AUC of the six graphs increased to an average ROC AUC of 83% points.

6 Summary and Conclusion

We investigated the idea of retrieving causal graph information from soft interventions that affect multiple variables and thus cannot deliver distinct structure information for causal graph construction. For this purpose, we created the definition of path constraints. We demonstrated how this information helps improving the results of the well established PCMCI+ algorithm. The procedure was demonstrated on a running combustion engine simulation. The obtained path constraints made it possible to increase the average ROC AUC value of the PCMCI+ results from 73% points to 83% points. Thus, we deem path constraints helpful in improving causal structure learning results. In future work, we will investigate more complex application scenarios. Additionally, we want explore the use of temporal information to gain additional structure information.

References

1. Crossley, P., Cook, J.: A nonlinear engine model for drivetrain system development. In: International Conference on Control 1991. Control'91. pp. 921–925. IET (1991)
2. Eberhardt, F., Scheines, R.: Interventions and causal inference. Philos. Sci. **74**(5), 981–995 (2007)
3. Fisher, R.A.: Design of experiments. Br. Med. J. **1**(3923), 554 (1936)
4. Hoyer, P., Janzing, D., Mooij, J.M., Peters, J., Schölkopf, B.: Nonlinear causal discovery with additive noise models. In: Advances in Neural Information Processing Systems. vol. 21. Curran Associates, Inc. (2008)
5. Janzing, D., Budhathoki, K., Minorics, L., Blöbaum, P.: Causal structure based root cause analysis of outliers. arXiv e-prints pp. arXiv–1912 (2019)

6. Kuehnert, C., Bernard, T., Frey, C.: Causal structure learning in process engineering using bayes nets and soft interventions. In: 2011 9th IEEE International Conference on Industrial Informatics. pp. 69–74. IEEE (2011)
7. Mao, X., Shang, P.: Transfer entropy between multivariate time series. Commun. Nonlinear Sci. Numer. Simul. **47**, 338–347 (2017)
8. Moskwa, J.J., Hedrick, J.K.: Automotive engine modeling for real time control application. In: 1987 American Control Conference. pp. 341–346. IEEE (1987)
9. Rubin, D.B.: Causal inference using potential outcomes: design, modeling, decisions. JASA **100**(469), 322–331 (2005)
10. Runge, J.: Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In: Conference on Uncertainty in Artificial Intelligence. pp. 1388–1397 (2020)
11. Weeks, R.W., Moskwa, J.J.: Automotive engine modeling for real-time control using matlab/simulink. SAE transactions pp. 295–309 (1995). https://www.mathworks.com/help/simulink/slref/modeling-engine-timing-using-triggered-subsystems.html
12. Woodward, J.: Making things happen: a theory of causal explanation. Oxford university press (2005)
13. Yeh, C.C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., Silva, D.F., Mueen, A., Keogh, E.: Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 1317–1322. Ieee (2016)

Reinforcement Learning from Human Feedback for Cyber-Physical Systems: On the Potential of Self-Supervised Pretraining

Timo Kaufmann, Viktor Bengs and Eyke Hüllermeier

Abstract

In this paper, we advocate for the potential of reinforcement learning from human feedback (RLHF) with self-supervised pretraining to increase the viability of reinforcement learning (RL) for real-world tasks, especially in the context of cyber-physical systems (CPS). We identify potential benefits of self-supervised pretraining in terms of the query sample complexity, safety, robustness, reward exploration and transfer. We believe that exploiting these benefits, combined with the generally improving sample efficiency of RL, will likely enable RL and RLHF to play an increasing role in CPS in the future.

Keywords

Reinforcement Learning from Human Feedback • Preference-Based RL • Self-Supervised Pretraining

1 Introduction

Reinforcement learning (RL) considers the setting of learning behavior from rewarded interaction with an environment. The reward function specifies the desired behavior while the

T. Kaufmann (✉) · V. Bengs · E. Hüllermeier
LMU Munich, Munich, Germany
e-mail: timo.kaufmann@ifi.lmu.de

V. Bengs
e-mail: viktor.bengs@ifi.lmu.de

E. Hüllermeier
e-mail: eyke@ifi.lmu.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_2

environment specifies the task dynamics. This setting is well-suited for cyber-physical systems (CPS), where the system repeatedly interacts with an environment to achieve some goal. RL can be used in this setting to learn a controller for a cyber-physical system, i.e., a policy that can choose appropriate actions based on the system's inputs. Examples of RL for CPS include applications to smart grids [18], HVAC [32], energy storage [31], autonomous driving [3], as well as legged robots [39, 43] and robotic manipulation [36].

One of the main challenges of applying RL to any task is measuring the agent's task performance in a way that is suitable for use as a reward function (reward design). Many of the largest successes of RL, such as as reaching or even exceeding human performance in the game of Go [37] and many Atari games [25], have been in the domain of games which have goals that are well-defined and easy to evaluate.

This is not the case for most real-world tasks however. Goals are often vague, subjective and characterized by trade-offs. Misspecifying these objectives can lead to surprising behaviors as well as safety issues [2]. Knox et al. [13] studies the challenges of reward design for autonomous driving, where the objective is a mixture of objective factors such as time to destination, fuel consumption and safety as well as subjective factors such as passenger experience. The right balance of these components may depend on context, such as time of day or the passenger's mood. More generally, Dulac-Arnold et al. [8] identifies reward design as one of the key challenges of applying RL to the real world.

RLHF is one way to cope with the challenge of reward design. Instead of assuming that a reward function is part of the problem specification, RLHF treats the reward function as part of the problem itself and attempts to learn it from human feedback. This is commonly done by collecting pairwise preference feedback over alternative agent trajectories (PbRL [42]) and using it to infer a reward function, but other feedback modalities such as (imperfect) demonstrations [11], corrections [20], critiques [7] or natural language [41] may be used as well.

Examples of RLHF include ChatGPT [28], an instance of a large language model fine-tuned with RLHF to follow instructions [29] in a dialogue context. Other examples from the language domain are summarization [40] and question answering [27]. Beyond text, RLHF has been used to guide image generation [12]. RLHF has also been used in games [6] as well as simulated continuous control tasks [6, 17]. In the domain of CPS, existing applications of RLHF include robot-to-human object handover [14] and robotic manipulation [5, 38].

RLHF can greatly reduce the challenge of reward design by enabling us to learn tasks that humans can judge, even if they are difficult to express in an engineered reward function. This avoids the need to explicitly specify all objectives or their trade-offs—those can be communicated by example instead. The reward model can be trained to estimate human preferences directly from the system's sensor inputs. If the sensor inputs convey sufficient information, the agent can even learn different trade-offs for different contexts. For example, an internal camera in an autonomous vehicle could be used to judge the mood of the passenger or detect the presence of a child and adapt the driving behavior accordingly.

2 The Potential of Pretraining

Learning rewards directly from sensor inputs presents us with a new challenge however, since these sensor inputs (especially when they are vision-based) are often high-dimensional. High-dimensional state- and action spaces are already a challenge for RL without human feedback [8]. In that setting the problem is often tackled by data augmentation [44], representation learning [15, 34] or model-based RL [10].

The latter two approaches—representation learning and model-based RL—can be considered instances of self-supervised learning [16, 22], a form of learning that tries to learn something about the structure of the input data from unlabeled examples. This can be achieved by generating labels from the input data itself, such as training models to predict hidden parts of the input data or to determine whether two data points are related (e.g., transformations of each other) or not. Self-supervised learning is commonly used to learn representations or to initialize networks which are then later fine-tuned to specific tasks. Since self-supervised learning does not require any explicit human labels, it is possible to train on large amounts of data. This has been an important driving factor behind recent successes in the domain of language models [4].

In model-based RL, the self-supervised objective is to predict the environment dynamics, i.e., predict the next state from the current state and a chosen action. The goal of state-representation learning is to learn a representation of the agent's state that makes downstream tasks, such as reward prediction or policy learning, easier. Consider the example of an agent tasked with controlling an autonomous car: While the raw state of an agent may consist of low-level sensor inputs such as the pixels captured by a camera, the learned representation should capture information that is immediately relevant to the driving task such as the car's position relative to other cars and pedestrians in a higher-level format. Such a representation can be learned from data that is already available, such as experiences of the environment dynamics [34], and can then enable more sample-efficient learning of the downstream task, such as reward prediction. See the overview by Lesort et al. [19] for a more detailed introduction to state representation learning.

In this paper, we want to highlight the potential of self-supervised pretraining in the form of state representation learning and world model learning to effectively learn behavior from human feedback. We expect pretraining can improve query sample complexity as well as the learning system's safety and robustness, allow for better exploration of the reward function and enable transfer of knowledge between tasks.

Query sample complexity: Starting with a good state representation has the potential to learn more accurate reward models while requiring fewer human labels. Such a representation can be learned in a self-supervised manner from unlabeled interactions with the environment [34] or as a side-effect of model-based RL [10, 26]. The learned representation is often more compact than the original observation and may also integrate information over multiple time-steps. This can be particularly beneficial in environments with high-dimensional observations such as images captured by a camera.

Similar sample-complexity benefits have been observed in RL without human feedback [34, 45], where learned state representations can often decrease the necessary amount of interaction with the environment or even enable the application of RL to domains in which it was previously not feasible.

Metcalf et al. [24] explores this idea for RLHF and observes that by encoding environment dynamics in the state representation, i.e., choosing the representation learning task in such a way that the representation of the next state can be predicted from the current one with a simple linear layer, results in a significant increase in sample efficiency.

In addition to explicit representation learning, sample efficiency could also be improved through data augmentation [30] as well as semi-supervised learning [30].

Safety: Instead of learning a state representation in isolation, it is also possible to learn a full model of the environment dynamics (world model). A world model provides the option of synthesizing queries, i.e., generating hypothetical behavior for feedback. This changes the active learning setting from (repeated) pool-based sampling to membership query synthesis [1]. Since these trajectories can be tailored to be informative about the human preferences, this can increase the sample efficiency of the preference learning process. In addition, synthesizing queries can increase the safety of the learning process since potentially dangerous behavior can be tested without actually performing it in the real world. Needless to say that this is particularly important when working with physical systems. Initial work has explored the potential of synthesized queries in an RLHF context [23, 33].

Another safety benefit of model-based RL is that it allows us to deploy separate policies in reality and in "imagination". Imagination refers to training that uses only interactions with the learned world model, not with the real environment. While the imagination policy may be focused on exploration, the real world policy may be focused on conservative data gathering.

Robustness: Synthesizing hypothetical behavior for feedback can not only improve the system's safety, but may also contribute to the robustness and generalization of the learned rewards. This is because the synthesized queries can explore edge-cases that would rarely be encountered in the pool of experiences. It is possible to actively optimize the queries to fill gaps in the agent's knowledge of the human preferences. The benefits of membership query synthesis over pool-based active learning are discussed by Elreedy et al. [9].

Reward exploration: Model-based RL can be used to improve the exploration behavior of RL agents by learning an exploration policy that leads the agent to novel states purely in imagination, which can then be deployed in the real environment for efficient exploration. This avoids the issue of retrospective novelty, where RL agents with intrinsic exploration bonuses optimize their policy to visit states which they previously found novel—which, by definition, they are not anymore once they are included in the training data.

This approach has successfully been applied for regular state-space exploration [35]. Since reward-space exploration can be similarly important as state-space exploration for RLHF [21], one might expect additional benefits by applying this principle to reward-space exploration as well.

Transfer: Yet another benefit of representation- and model-learning is the possibility of transferring knowledge between tasks. Since a world model or state representation that was learned for one task remains valid for any other task with the same dynamics, this knowledge can be transferred and reward models for new tasks can be learned faster. A similar effect for model-based RL without human feedback is discussed by Moerland et al. [26].

3 Discussion and Conclusion

Learning controllers for cyber-physical systems has the potential of enabling many new use cases with complex interactions and increased integration of multiple systems. This may be of use for many applications, such as robotics, smart buildings and autonomous vehicles.

While to date applications of RL to real-world systems are sparse, the increasing sample efficiency of RL combined with the increased applicability to many tasks thanks to RLHF may cause that to change in the near future. Improving the feedback-efficiency of RLHF with approaches such as the ones discussed in this paper is therefore a promising area of future research. We believe that self-supervised pretraining has many benefits to offer and could play a crucial part in opening up many new use cases for cyber-physical systems.

Acknowledgements This publication was supported by LMUexcellent, funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder as well as by the Hightech Agenda Bavaria.

References

1. Aggarwal, C.C., Kong, X., Gu, Q., Han, J., Yu, P.S.: Active learning: a survey. In: Data Classification: algorithms and Applications. CRC Press (2014). https://doi.org/10.1201/b17320-23
2. Amodei, D., Olah, C., Steinhardt, J., Christiano, P.F., Schulman, J., Mané, D.: Concrete problems in AI safety. CoRR **abs/1606.06565** (2016). http://arxiv.org/abs/1606.06565
3. Bai, Z., Shangguan, W., Cai, B., Chai, L.: Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic. 2019 Chinese Control Conference (CCC) (2019)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei,

D.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems (2020)

5. Cabi, S., Colmenarejo, S.G., Novikov, A., Konyushova, K., Reed, S., Jeong, R., Zolna, K., Aytar, Y., Budden, D., Vecerik, M., Sushkov, O., Barker, D., Scholz, J., Denil, M., de Freitas, N., Wang, Z.: Scaling data-driven robotics with reward sketching and batch reinforcement learning. In: Proceedings of Robotics: Science and Systems (2020). https://doi.org/10.15607/RSS.2020.XVI.076
6. Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Advances in Neural Information Processing Systems (2017)
7. Cui, Y., Niekum, S.: Active reward learning from critiques. In: 2018 IEEE International Conference on Robotics and Automation, ICRA (2018). https://doi.org/10.1109/ICRA.2018.8460854
8. Dulac-Arnold, G., Levine, N., Mankowitz, D.J., Li, J., Paduraru, C., Gowal, S., Hester, T.: Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Machine Learning **110**(9) (2021). https://doi.org/10.1007/s10994-021-05961-4
9. Elreedy, D., Atiya, A.F., Shaheen, S.I.: A novel active learning regression framework for balancing the exploration-exploitation trade-off. Entropy **21**(7) (2019). https://doi.org/10.3390/e21070651
10. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.P.: Mastering diverse domains through world models. CoRR **abs/2301.04104** (2023). https://doi.org/10.48550/arXiv.2301.04104
11. Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., Amodei, D.: Reward learning from human preferences and demonstrations in Atari. In: Advances in Neural Information Processing Systems (2018)
12. Kazemi, H., Taherkhani, F., Nasrabadi, N.M.: Preference-based image generation. In: IEEE Winter Conference on Applications of Computer Vision, WACV 2020 (2020). https://doi.org/10.1109/WACV45572.2020.9093406
13. Knox, W.B., Allievi, A., Banzhaf, H., Schmitt, F., Stone, P.: Reward (mis)design for autonomous driving. CoRR **abs/2104.13906** (2021). https://arxiv.org/abs/2104.13906
14. Kupcsik, A.G., Hsu, D., Lee, W.S.: Learning dynamic robot-to-human object handover from human feedback. CoRR **abs/1603.06390** (2016). http://arxiv.org/abs/1603.06390
15. Laskin, M., Srinivas, A., Abbeel, P.: CURL: Contrastive unsupervised representations for reinforcement learning. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. vol. 119. PMLR (2020)
16. LeCun, Y., Misra, I.: Self-supervised learning: The dark matter of intelligence. Meta AI (2021). https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/, accessed: 2023-01-26
17. Lee, K., Smith, L.M., Abbeel, P.: PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021. vol. 139. PMLR (2021)
18. Lei, L., Tan, Y., Dahlenburg, G., Xiang, W., Zheng, K.: Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids. IEEE Internet of Things J. **8** (2020)
19. Lesort, T., Díaz-Rodríguez, N., Goudou, J.F., Filliat, D.: State representation learning for control: an overview. Neural Networks **108**, 379–392 (2018). https://doi.org/10.1016/j.neunet.2018.07.006
20. Li, M., Canberk, A., Losey, D.P., Sadigh, D.: Learning human objectives from sequences of physical corrections. In: IEEE International Conference on Robotics and Automation, ICRA (2021). https://doi.org/10.1109/ICRA48506.2021.9560829
21. Liang, X., Shu, K., Lee, K., Abbeel, P.: Reward uncertainty for exploration in preference-based reinforcement learning. In: International Conference on Learning Representations (2022)

22. Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J.: Self-supervised learning: Generative or contrastive. IEEE Trans. Knowl. Data Eng. **35**(1) (2023). https://doi.org/10.1109/TKDE.2021.3090866
23. Liu, Y., Datta, G., Novoseller, E.R., Brown, D.S.: Efficient preference-based reinforcement learning using learned dynamics models. CoRR **abs/2301.04741** (2023). https://doi.org/10.48550/arXiv.2301.04741
24. Metcalf, K., Sarabia, M., Theobald, B.: Rewards encoding environment dynamics improves preference-based reinforcement learning. CoRR **abs/2211.06527** (2022). https://doi.org/10.48550/arXiv.2211.06527
25. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540) (2015). https://doi.org/0.1038/nature14236
26. Moerland, T.M., Broekens, J., Plaat, A., Jonker, C.M.: Model-based reinforcement learning: a survey. Found. Trends Mach. Learn. **16**(1), 1–118 (2023). https://doi.org/10.1561/2200000086
27. Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., Schulman, J.: WebGPT: browser-assisted question-answering with human feedback. CoRR **abs/2112.09332** (2021), https://arxiv.org/abs/2112.09332
28. OpenAI: ChatGPT: Optimizing language models for dialogue (2022). https://openai.com/blog/chatgpt/. Accessed: 23. Jan. 2023
29. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: Advances in Neural Information Processing Systems (2022)
30. Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., Lee, K.: SURF: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In: International Conference on Learning Representations (2022)
31. Qi, B., Rashedi, M., Ardakanian, O.: EnergyBoost: Learning-based control of home batteries. In: Proceedings of the Tenth ACM International Conference on Future Energy Systems, e-Energy 2019 (2019). https://doi.org/10.1145/3307772.3328279
32. Raman, N.S., Devraj, A.M., Barooah, P., Meyn, S.P.: Reinforcement learning for control of building HVAC systems. In: 2020 American Control Conference, ACC 2020. IEEE (2020). https://doi.org/10.23919/ACC45564.2020.9147629
33. Reddy, S., Dragan, A.D., Levine, S., Legg, S., Leike, J.: Learning human objectives by evaluating hypothetical behavior. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. vol. 119. PMLR (2020)
34. Schwarzer, M., Anand, A., Goel, R., Hjelm, R.D., Courville, A.C., Bachman, P.: Data-efficient reinforcement learning with self-predictive representations. In: International Conference on Learning Representations (2021)
35. Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., Pathak, D.: Planning to explore via self-supervised world models. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. vol. 119. PMLR (2020)
36. Sermanet, P., Xu, K., Levine, S.: Unsupervised perceptual rewards for imitation learning. In: International Conference on Learning Representations (2017)
37. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T.P., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nature **529** (2016)

38. Singh, A., Yang, L., Finn, C., Levine, S.: End-to-end robotic reinforcement learning without reward engineering. In: Robotics: Science and Systems XV (2019). https://doi.org/10.15607/RSS.2019.XV.073
39. Smith, L.M., Kostrikov, I., Levine, S.: A walk in the park: learning to walk in 20 minutes with model-free reinforcement learning. CoRR **abs/2208.07860** (2022). https://doi.org/10.48550/arXiv.2208.07860
40. Stiennon, N., Ouyang, L., Wu, J., Ziegler, D.M., Lowe, R., Voss, C., Radford, A., Amodei, D., Christiano, P.F.: Learning to summarize from human feedback. CoRR **abs/2009.01325** (2020), https://arxiv.org/abs/2009.01325
41. Williams, E.C., Gopalan, N., Rhee, M., Tellex, S.: Learning to parse natural language to grounded reward functions with weak supervision. In: 2018 IEEE International Conference on Robotics and Automation, ICRA (2018). https://doi.org/10.1109/ICRA.2018.8460937
42. Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J.: A survey of preference-based reinforcement learning methods. J. Mach. Learn. Res. **18** (2017). http://jmlr.org/papers/v18/16-634.html
43. Wu, P., Escontrela, A., Hafner, D., Goldberg, K., Abbeel, P.: DayDreamer: world models for physical robot learning. CoRR **abs/2206.14176** (2022). https://doi.org/10.48550/arXiv.2206.14176
44. Yarats, D., Kostrikov, I., Fergus, R.: Image augmentation is all you need: regularizing deep reinforcement learning from pixels. In: International Conference on Learning Representations (2021)
45. Yu, T., Lan, C., Zeng, W., Feng, M., Zhang, Z., Chen, Z.: PlayVirtual: augmenting cycle-consistent virtual trajectories for reinforcement learning. In: Advances in Neural Information Processing Systems (2021)

Using ML-Based Models in Simulation of CPPSs: A Case Study of Smart Meter Production

Nemanja Hranisavljevic, Tom Westermann, Philip Kroke
and Carsten Waschkies

Abstract

Simulation models have proven successful in various CPPS tasks such as optimization, diagnosis or reconfiguration. However, creating these models is a costly process. This paper describes an approach which uses: 1) recorded data to automatically learn timed automata models of system components; and 2) manual logic based on prior knowledge that extends and enables the utilization of the learned models for simulation. Experiments in a smart meter production facility show: 1) a successful detection of a suboptimal configuration; 2) the identification of causes of a decrease in productivity; and 3) a correct assessment of possible actions after a disturbance has occurred.

Keywords

CPPSs • Timed Automata • Simulation • Machine Learning

N. Hranisavljevic (✉) · T. Westermann
Institute of Automation Technology, Helmut Schmidt University, Hamburg, Germany
e-mail: hranisan@hsu-hh.de

T. Westermann
e-mail: tom.westermann@hsu-hh.de

P. Kroke · C. Waschkies
eBZ GmbH, Bielefeld, Germany
e-mail: carsten.waschkies@ebzgmbh.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien
für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_3

1 Introduction and Problem Statement

As defined by the Industry 4.0 agenda, modern production systems (often called Cyber-Physical Production Systems or CPPSs) rely on intelligent services such as self-optimization, self-reconfiguration and self-diagnosis [6, 8, 10] which should increase the robustness, flexibility and productivity of the systems.

CPPSs consist of many autonomous and cooperative subsystems and components that interact in a multitude of ways. This combination of individual subsystems can cause emergent behavior, i.e. behavior of the full system that deviates from the expected behavior of the subsystems [9]. This often prevents both creating the holistic behavior model at design time and straightforward updating after changes are conducted in the system.

One possibility to handle this problem lies in data-driven approaches, which allow the learning of an accurate model from the actual observed data during the plant operation. In these approaches, abstractions of the subsystem's behavior are derived from the machine data.

Due to the high variability of CPPSs, the creation/learning of these models is often not physics-based, but takes a more generic form like timed automata or Petri-nets. Nevertheless, these models can be very difficult to learn for complex systems, especially on a level that would enable the simulation of the system behavior. On the other hand, the simulation of different scenarios and decisions could point to suboptimal behavior in some of the components, while the online observations can be compared to the model and errors can be detected [10].

More precisely, the models can be used in three modes [8] (see Fig. 1):

1. *Offline Simulation*—evaluating system configurations for different contexts (e.g. new product type, new machine module) [5][17]
2. *Proactive Simulation*—recognizing deviations from the expected behavior given the current context [10][3]
3. *Reactive Simulation*—evaluating a possible sequence of actions after a disturbance has occurred [12][2]

This paper proposes an approach validated on a real-world industrial use case, in which ensembles of timed automata simulation models were learned from data traces generated in the manufacturing of smart meters. These models were then used in three scenarios spanning all three of the aforementioned simulation modes.

(RQ1) What part of the simulation model can be learned (ML) and what part needs to be manually-programmed based on collected prior knowledge?

(RQ1) How can the integration between the manually-programmed and the data-driven/learned part be achieved?

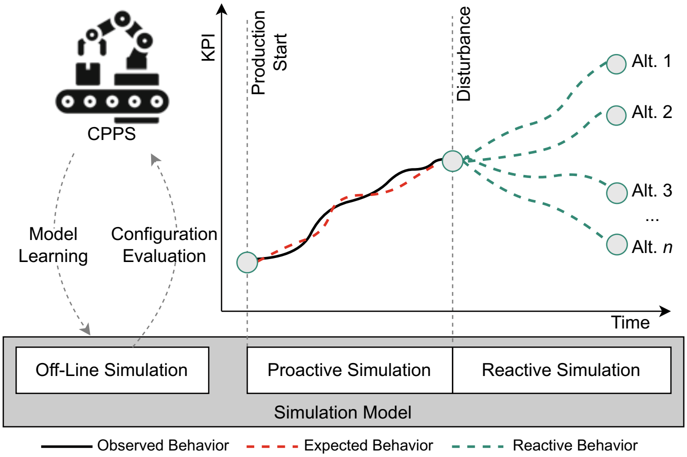


Fig. 1 A learned model can be used for evaluating a configuration, monitoring the CPPSs behavior, and evaluating decision options in case of a disturbance

2 Use Case

In this work, a use case of a company called eBZ GmbH[1] from Bielefeld, Germany is used to evaluate our simulation approach based on ML models of dynamic system behavior. The plant of interest is responsible for the assembly, programming, and testing of smart meters. Figure 2 describes the plant behavior using the formalized process description (VDI 3682, [15]).

Order data The production is performed in an order fulfillment manner using an assemble-to-order policy [14], in which only one order can be active at a time. The order data is available from the manufacturing execution system (MES), each order having an order ID, a product type and the number of products. During production, for each order we calculate the productivity metric as the order's average throughput.

Event data The event logs are generated by the MES and contain information about the communication between the MES and the individual CPPS components. Each row in the tabular data gives the timestamp of the event, the event symbol (name), the component which logged the event, the ID of the product piece it has affected and the ID of the order that this piece belongs to.

Value-discrete data System PLCs provide access to a subset of program variables (e.g. via OPC-UA interface). Each row in the tabular data gives the time-stamp of the value change,

[1] www.ebzgmbh.de

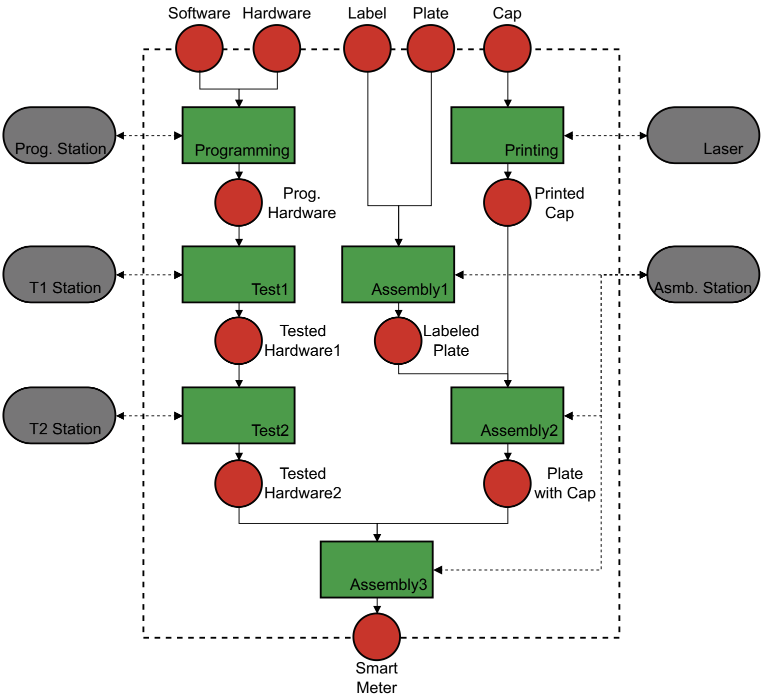


Fig. 2 Formalized Process Description of the smart meter production process. Each technical resource can have multiple instances (E.g. $2\times T1$, $8\times T2$...)

the name of the variable that changed its value, the new value and the component that this variable belongs to.

3 Proposed Approach

The proposed approach is based on discrete-event models of the plant components and relies on the well-established deterministic timed automata [1]. Here, “timed” refers to the continuous clock measuring time spent in some state until a transition occurs, while “deterministic” refers to the deterministic trajectory of states given a sequence of events applied to the automaton.

Model It extends the standard timed automaton by a probabilistic aspect, which refers to the categorical probability distributions (or relative frequency) of events given each state (see Fig. 3). The result is a generative model which can be used to simulate system behavior starting from any initial state. Its single continuous clock t is reset on every event. To generate

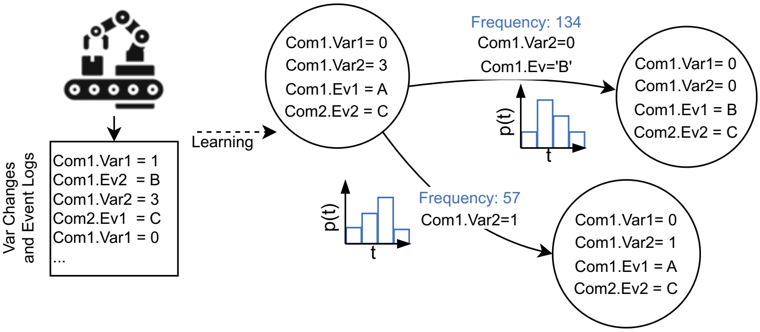


Fig. 3 The proposed timed-automata learning approach

the next event, first an event symbol e is sampled given the current state q, then the clock timing of the event is sampled according to the one-dimensional distribution $p(t|q, e)$.

Learning algorithm As shown in previous works, timed automata can be learned efficiently from observations [7, 11]. In this work we use a simple learning approach applied to subsets of variables/events—these subsets are partitioned according to prior knowledge about related components/variables/events. The algorithm assumes that the system state at any moment is fully observed and determined by: 1) the observed values of discrete variables; and 2) the last logged event symbol of each considered component (see Fig. 3). After the dataset is iterated, the probability distributions of event clock timings can be approximated by means of histograms.

Manual logic While the learned automata approximate the behavior of the system components, additional logic is necessary to coordinate the behavior of the complete automata ensemble. This article proposes a *Manual Logic* component (see Fig. 4) able to forward the events that are generated by any of the automata to a group of other automata, in order to trigger a transition there. Additionally, it allows changing the plant configuration i.e. the transition parameters of the models. The *Manual Logic* is manually programmed based on the collected prior knowledge.

Offline Simulation Here, multiple plant configurations are simulated and the performance that they achieve for a given order is evaluated. Therefore the approach takes a search space of plant configurations as well as order data as input, and simulates the automata. Therefore the *Manual Logic* and the automata are fixed, while the configuration is searched.

Proactive Simulation In the proactive simulation, the learned automata are used to compare the online data to the simulation. The plant configuration and the *Manual Logic* are fixed. If the ongoing productivity is lower than the simulated productivity, this hints at faults in

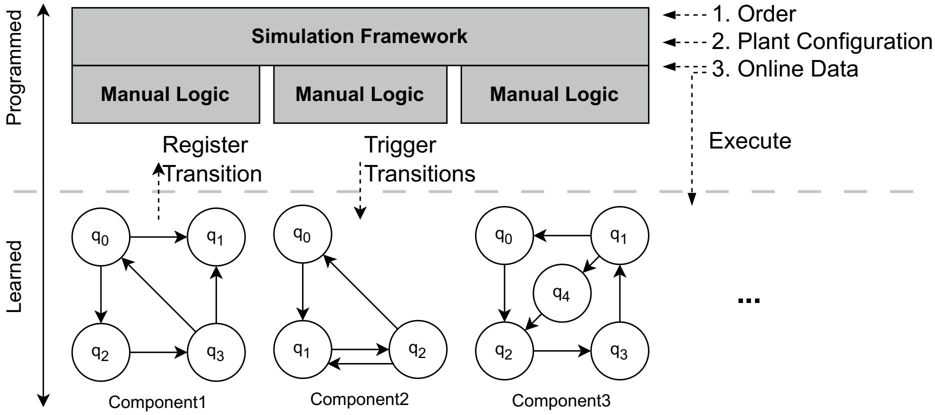


Fig. 4 *Manual Logic* component is the key to enabling ML-based simulation

components or problems with the used control logic. For example, if a lower productivity is observed in the online data, it could be identified that a robot is working more slowly than what was learned in the automata. By comparing the online data to a new simulation run based on the corrected robot timings, we can confirm that this was the probable cause of the problem.

Reactive Simulation Here, the plant configuration is fixed and the automata only execute the online data of the current order. Once a disturbance has been detected, the simulations can be started using the current automata states and clock values. Given a set of possible decisions that can be made after the disturbance, the *Manual Logic* component runs many simulations trying different alternatives, in order to determine a sequence of actions that should lead to the highest system performance. For example, in some systems, a robot might have two alternatives: going for a new hardware piece and putting it on a conveyor or assembling the casing of another half-finished product.

4 Experiments

Data set The data set consists of 3.25 million rows of discrete data and 0.7 million rows of event data during the seven days of production. A total of 207 discrete variables are observed, some of which are changing rapidly, while others are changing the value only a few times a day, The data set and the part of the used code are available on Kaggle[2].

[2] www.kaggle.com/datasets/nemanjahrane/ebz-plant-2a-2b

Learning Model learning was implemented in Python using the algorithm described in Sect. 3. Manually-collected prior knowledge was used to decide which events/components should be jointly learned. This led to seven learned automata models, two of which are presented in Fig. 5.

Manual logic It is programmed manually based on the prior knowledge and the information given in the process description (on Fig. 2). Some further constraints were added based on process knowledge, e.g. that all *T1 (Test1)* and *T2 (Test2)* components are always triggered simultaneously.

Offline Simulation There are various configuration parameters that can be analyzed in this plant. The following scenario was considered:

> The number of *T2* components can be changed. Given a specific product type, what is the optimal number N_{T2} of *T2* components to use?

In order to answer the question, a search space was defined for the parameter $N_{T2} \in \{2, ..., 11\}$ (due to the technical constraints). The simulation is then performed for each of the 10 options. The results on the right side of Fig. 6 show that the optimal N_{T2} depends of the product type. For one product type two $T2$ components are enough (grey), while for another product type productivity increases taper off after the sixth component is added (blue). These results were later confirmed at the real CPPS.

Proactive Simulation Here, productivity was chosen as the performance indicator which is used to compare the simulated and the actual behavior of the plant. Consider the following scenario:

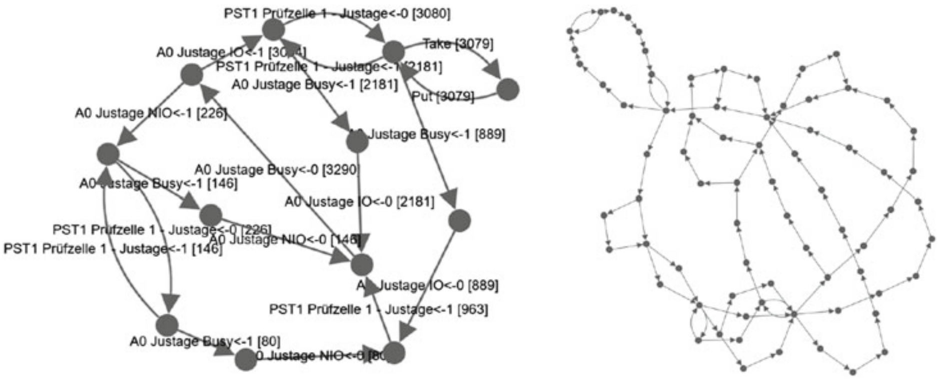


Fig. 5 Learned automata of the *Programming* (*Left*) and *Laser* (*Right*) components

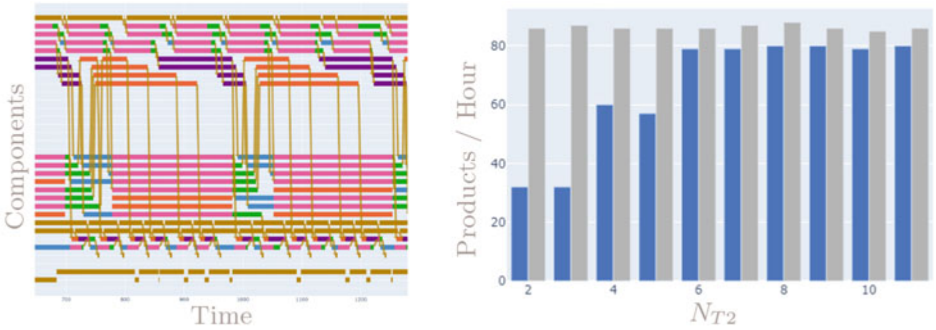


Fig. 6 *Left:* Simulated behavior as a Gantt chart. Arrows indicate product transports across components, while colors represent different component states. *Right:* Offline analysis of order productivity sensitivity to the N_{T2} parameter for two product types (gray/blue bars)

> A faulty *T2* component leads to products being wrongly marked as defective which causes a drop in productivity. How can we detect and identify this problem?

The left side of Fig. 7 shows a diagram where—according to the simulation—around 70 products should have been produced in an hour, while the real CPPS only produced 44. The simulation allows us to determine when this slowdown occurred, as well as to investigate which events possibly occurred with a delay, or too frequently. A further analysis leads to the explanation of the problem: too many *Failed* events in one of the *T2* stations.

Reactive Simulation In general, various disturbances might occur in the plant. Here, the following scenario was investigated:

> A number of *Defect* products occur in one of the three processes: *Programming*, *T1* or *T2*. Which sequence of possible decisions results in the least loss in productivity?

First, the disturbance is found on 9th of January at 8 am. Then, sequences of decision alternatives for the different robot tasks are simulated starting from the selected point in time.

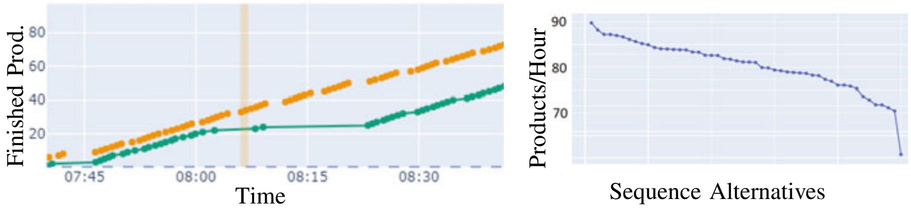


Fig. 7 *Left:* Proactive-simulated productivity (orange) and actual productivity (green). The yellow bar marks a disturbance. *Right:* Sorted hourly productivity of the simulated reactive alternatives

The right side of Fig. 7 shows a significant difference in the evaluated decision sequences, of which the best one should be chosen.

5 Conclusions and Future Work

In this paper we presented an approach to the simulation of CPPS behavior based on: 1) timed automata that were learned from the data; and 2) manual logic programmed using the collected prior knowledge. The approach was successfully validated in three scenarios:

1. Using offline simulation to optimize the number of test stations to achieve a higher productivity than the base configuration;
2. Evaluating the current performance against the theoretical performance using proactive simulation;
3. Adapting the decision sequence of a robot after a series of defective products using reactive simulation.

However, the approach also has the drawback of being very labor intensive to set up. Large manual effort was needed to determine the set of events that are relevant and determine their meaning with regards to the automaton. While this information can be considered prior knowledge, it is often not readily available.

Future research could focus on representing this information in the form of ontologies and knowledge graphs [16] to achieve a more detailed highly formalized representation of knowledge. The field of Informed ML offers numerous approaches to further integrate this knowledge in the ML process [13]. Additionally, incorporating possibly available continuous data can also be considered [4].

Acknowledgements This work has been partially supported and funded by the German Federal Ministry of Education and Research (BMBF) for the project "Time4CPS - A Software Framework for the Analysis of Timing Behavior of Production and Logistics Processes" under the contract number 01IS20002. It was partially developed within the Fraunhofer Cluster of Excellence "Cognitive Internet Technologies". The authors thank Witalij Gamerman from eBZ GmbH for meaningful discussions about the analyzed use case.

References

1. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theor. Comput. Sci. **126**(2), 183–235 (4 1994). https://doi.org/10.1016/0304-3975(94)90010-8
2. Balzereit, K., Niggemann, O.: AutoConf: A New Algorithm for Reconfiguration of Cyber-Physical Production Systems. IEEE Transactions on Industrial Informatics (2022)

3. Diedrich, A., Maier, A., Niggemann, O.: Model-based Diagnosis of Hybrid Systems using Satisfiability Modulo Theory. Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19) (2019)
4. Hranisavljevic, N., Maier, A., Niggemann, O.: Discretization of hybrid CPPS data into timed automaton using restricted Boltzmann machines. Eng. Appl. Artif. Intell. **95**, 103826 (2020)
5. Ladiges, J., Fülber, A., Arroyo, E., Fay, A., Haubeck, C., Lamersdorf, W.: Learning material flow models for manufacturing plants from data traces. In: Proceeding – 2015 IEEE International Conference on Industrial Informatics, INDIN 2015. pp. 294–301. Institute of Electrical and Electronics Engineers Inc. (9 2015)
6. Lee, J., Davari, H., Singh, J., Pandhare, V.: Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. Manufacturing Letters (2018)
7. Maier, A.: Online Passive Learning of Timed Automata for Cyber-Physical Production Systems. In: The 12th IEEE International Conference on Industrial Informatics (INDIN 2014). Porto Alegre, Brazil (7 2014)
8. Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K.: Cyber-physical systems in manufacturing. CIRP Annals – Manuf. Technol. **65**, 621–641 (2016)
9. Monostori, L.: Cyber-physical production systems: roots, expectations and R&D challenges. PROCEDIA CIRP **17**, 9–13 (2014)
10. Niggemann, O., Lohweg, V.: On the Diagnosis of Cyber-Physical Production Systems – State-of-the-Art and Research Agenda. In: Twenty-Ninth Conference on Artificial Intelligence (AAAI-15). Austin, Texas, USA (2015)
11. Niggemann, O., Stein, B., Vodenčarević, A., Maier, A., Kleine Büning, H.: Learning Behavior Models for Hybrid Timed Systems. In: Twenty-Sixth Conference on Artificial Intelligence (AAAI-12). pp. 1083–1090. Toronto, Ontario, Canada (2012)
12. Pfrommer, J., Zimmerling, C., Liu, J., Kärger, L., Henning, F., Beyerer, J.: Optimisation of manufacturing process parameters using deep neural networks as surrogate models. Procedia CIRP **72**, 426–431 (2018)
13. von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Walczak, M., Pfrommer, J., Pick, A., Ramamurthy, R., Garcke, J., Bauckhage, C., Schuecker, J.: Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. IEEE Transactions on Knowledge and Data Engineering pp. 1–20 (2021)
14. Song, J.S., Zipkin, P.: Supply Chain Operations: Assemble-to-Order Systems. In: Supply Chain Management: Design, Coordination and Operation, Handbooks in Operations Research and Management Science, vol. 11, pp. 561–596. Elsevier (2003)
15. VDI/VDE—Verein Deutscher Ingenieure e.V./Verband der Elektrotechnik Elektronik Informationstechnik: VDI/VDE 3682-1: Formalised process descriptions – Concept and graphic representation. Tech. rep., Verein Deutscher Ingenieure, Düsseldorf, D (2015)
16. Westermann, T., Hranisavljevic, N., Fay, A.: Accessing and interpreting opc ua event traces based on semantic process descriptions. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–7 (2022). https://doi.org/10.1109/ETFA52439.2022.9921565
17. Windmann, S., Niggemann, O.: Automatic model separation and application to diagnosis in industrial automation systems. In: IEEE International Conference on Industrial Technology (ICIT 2015) (2015)

Deploying Machine Learning in High Pressure Resin Transfer Molding and Part Post Processing: A Case Study

Jasper Steffens, Robin Kühnast-Benedikt, Florian Leber, Philipp Rosenberg, and Frank Henning

Abstract

High pressure resin transfer molding (HP-RTM) is well suited to medium volume automated production of composites. The process complexities of HP-RTM however often make its application insular. Data is not carried forward along the production chain and process reliability is assessed as a unified indicator with minimal granular consideration of individual contributing factors. Cause and effect relationships spanning the process chain remain undetected. Predator (10/2020–09/2023) is an ongoing Eurostars

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-47062-2_4.

J. Steffens (✉) · P. Rosenberg · F. Henning
Department of Polymer Engineering, Fraunhofer Institute for Chemical Technology (ICT), Pfinztal, Germany
e-mail: jasper.steffens@ict.fraunhofer.de

P. Rosenberg
e-mail: philipp.rosenberg@ict.fraunhofer.de

F. Henning
e-mail: frank.henning@ict.fraunhofer.de

R. Kühnast-Benedikt · F. Leber
Boom Software AG, Leibnitz, Austria
e-mail: r.kuehnast-benedikt@boomsoftware.com

F. Leber
e-mail: f.leber@boomsoftware.com

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_4

project aiming to bridge this divide by developing an intelligent data processing system across the industrial process chain of composite production. The consortium has already developed an approach to acquire and transfer meaningful process related data from molding to post-processing of parts. The data collection merges RTM tooling, equipment sensors, structure-borne sound data and tool wear measurements during the milling process. Unique part identifiers allow traceability of production parameters for online quality assurance and data-based optimization across the process chain. The developed approach enables tool wear monitoring as well as tailored predictive maintenance and enhanced remote customer support in addition to a data-driven understanding of the production process.

Keywords

HP-RTM • Milling • Digitalization • Data-acquisition • Data-analysis

1 Introduction

Competitiveness in the composite production market is dependent on robust processes to manufacture components of consistent quality in short cycle times. The required process understanding is currently retained as operator experience and other domain expert knowledge. Such expertise is typically insular, focused on individual steps along the production chain from raw material to finished part and not trivially available to inform data driven analysis [1, 2]. This hinders the discovery of latent variables and their complex interactions along the process chain. Data-driven business models such as predictive maintenance, remote support services for production equipment and advanced part quality assurance approaches require a more stringent collection, storage and evaluation of process information [3–5].

1.1 Composite Manufacturing by RTM

High pressure resin transfer molding (HP-RTM) is a method of combining fibers and polymers into fiber reinforced polymer composites [6]. As shown in Fig. 1, the process begins with cutting and assembling of a textile reinforcement into a stack. This material is then draped close to the final part shape. During this step, the stack is also consolidated, which stabilizes it for further handling. After shaping the stack, the preform is transferred to the RTM tool. The resin is injected under controlled pressure and temperature in this step, which polymerizes to the desired composite matrix material. Once this curing is complete, the part is removed from the tool and machined, typically milled, to meet the target dimensions.

The RTM process is affected by multiple factors such as effective fiber volume content, textile permeability, resin viscosity, tool gap, pressure, etc. In a classical approach, each

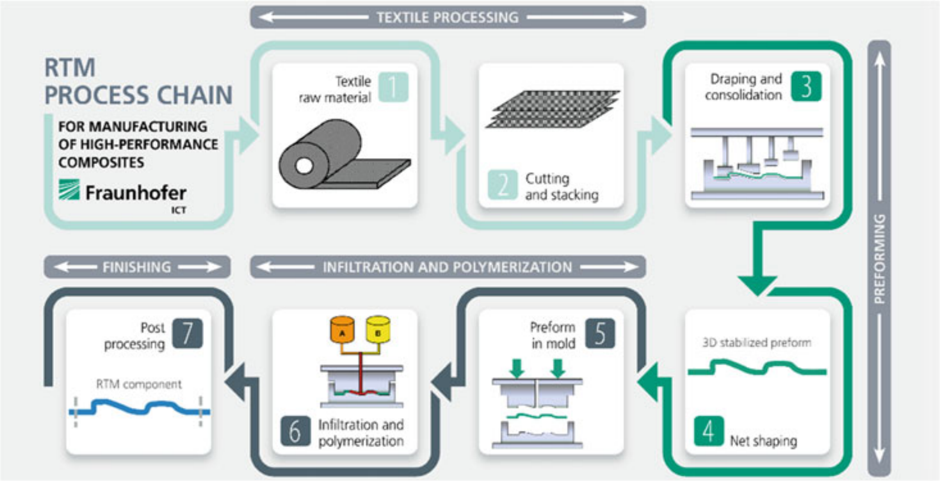


Fig. 1 RTM process chain for composite production, consisting of (1) textile raw material from fibers, (2) cutting and stacking of textile reinforcement, (3) draping and consolidation, (4) net shaping of the preform, (5) transfer of preform into the mold, (6) infiltration and polymerization of the polymer matrix and (7) post processing of the composite part

influencing variable is controlled for by fine tuning the process during tool commissioning and maintaining a conservative safety margin in all process parameters where possible.

Similarly, in the final post-processing step of milling the composites, tool wear is a critical factor. Tool damage accumulates over time and tool life is typically assumed conservatively to avoid reduced milling quality.

1.2 Knowledge Extraction in a Complex Network of Cyber-Physical Systems

Bridging the steps of composite production to create a unified, digital representation of the part production requires knowledge extraction at each step of the process.

The gain of non-trivial information from a complex network of digitized and connected assets along the value stream of a producing organization requires experts in the specific domain, IT, data science and management. The underlying reference process is widely studied in the literature [7–9] and the following steps were incorporated in the project:

1. Business understanding
2. Data understanding
3. Data preparation
4. Data modelling

5. Evaluation
6. Deployment

2 Implemented Approach

2.1 Data Management and Analysis

To allow a meaningful discussion of process data, three required stages of activities can be formulated [10]:

1. Identification of data sources, boundary conditions & use-cases
2. Definition of data transfer, storage and analysis methods
3. Implementation of the software and hardware modules to the manufacturing equipment

A successful implementation of these steps allows for the derivation of data analysis models for a process, in this case HP-RTM and milling.

As part of the first step, data was grouped by process step (RTM, milling) and origin (tool, press, RTM equipment, milling fixture, other). A list of possible faults in product or process with relevant equipment components and their data sources was generated in exchange with the respective domain experts for all process steps covered. Identifying these use-cases allowed for a derivation of requirements on the data collection modules. These cover the RTM process from infiltration to post processing (see Fig. 1, steps 3–7).

Stage one showed disparate requirements between the process steps RTM and milling, leading to the implementation of two tailored data acquisition solutions. In the case of RTM, a centralized data merging node must communicate with the tool, press and RTM equipment, which in turn were enabled to share their sensor data. Additional sensors were included where use-cases suggested a probable benefit. In the case of milling, an equipment-borne sensor was combined with offline measurements of tool wear.

To enable unique identifiability of experiments across both process steps, bar code identifiers were embedded into the parts. This consequently enabled merging of both the data sets into the project database for an integrated analysis of activities (Refer to Fig. 2 for a high-level overview of the resulting data acquisition structure).

2.2 Process Monitoring and Predictive Maintenance for serial HP-RTM Production

Process monitoring and maintenance support activities were identified as desired use-cases for RTM. This informed the tool design, incorporating technological features benefitting

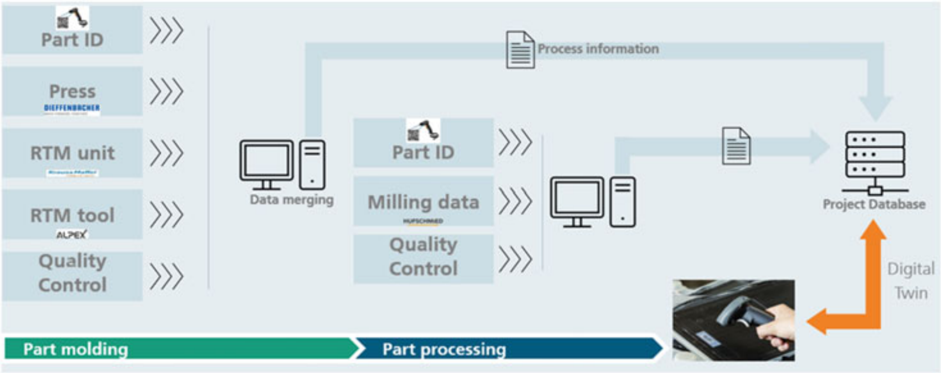


Fig. 2 Schematic view of the developed data acquisition structure. During part molding information from RTM equipment, tooling and press is captured together with quality control data. In part processing milling and quality control data is captured and merged. All data is referenced to individual part identifiers and saved in a common data store

most from these solutions. This includes complex sealing concepts and tool functionalities. The size and setup of the experimental campaigns was chosen to approach serial production conditions as closely as possible within the constraints of a research project. For this purpose, the material handling was automated once it was defined, implemented and tested.

2.3 Process Monitoring and Quality Assurance in Post-Processing

Process monitoring and quality assurance (QA) related use-cases were identified for post-processing (milling). Variability in milling tool design required a large number of experiments to allow for a comprehensive testing plan to be developed but also necessitated the development of a customized milling fixture.

Typically, milling operations in composite manufacture by RTM can be parallelized cost effectively—in contrast to the infusion step. The availability of a single research fixture thus limited the size of experimental campaigns for the project.

3 Preliminary Results

As of the writing of this paper, two experimental campaigns have been conducted, encompassing a total of 991 individually identifiable parts. Both data analysis and part processing activities are still ongoing. This section is, therefore, intended to showcase one exemplary finding, enabled by the approach developed. The first use case was prioritized using a criticality-centered approach that primarily focused quality and availability losses. Decisive for the importance of the respective losses was the impact and frequency in the production domain. In the first experimental campaign, the seal of the press was not maintained regularly. The maintenance strategy was reactive, leading to the full degradation of its condition. This allowed tracking of important information about the failure-behavior. In real-world scenarios, it is critical to collect data of negative incidents. This typically is at odds with the key objective of maintenance organizations to ensure reliability and availability.

3.1 Comparison of Physical to Date-Centric Modelling

The identification of rules that lead to the loss of quality or function is a crucial responsibility on the shopfloor. The task is usually performed with deep domain and engineering knowledge. In order to show the benefits related to the presented methodology, the following figure and table show the results of a conventional approach of incorporating physical modelling in comparison to a data-centric approach. The goal was to model the condition of the seal of the press.

The following models were trained in the first experimental campaign and tested on the second experimental campaign. The RMSE was calculated on the test data.

Linear Regression:
Despite its static nature, univariate linear regression is widely used in the industry. The data understanding and preparation effort is straightforward. The modelling is easy and the deployment doable. The disadvantage lies in the accuracy and robustness of the model. The R2 was 0.845 on the training data.

Non-Linear Regression:
The second frequently used technique to model and predict metric variables fast is by building a non-linear regression model. The advantages and disadvantages are similar to the linear regression model, with the additional benefit that many degradation curves follow a non-linear trend (compare Fig. 3). The R^2 was 0,96 on the training data.

Multiple Regression:
In comparison to the first two static regressions, the multiple regression includes multiple variables to determine and explain the behavior of the seal condition. Variables that detect influencing effects were identified and used to model the behavior of the seal-condition as measured by in-tool sensors. The accuracy increased significantly, as factors such as the specific experimental setup were included. The R2 was 0.97 on the training data.

The comparison of the three models is summarized in Table 1. The evaluation was performed in the second campaign. It is clearly shown that the first two models initially

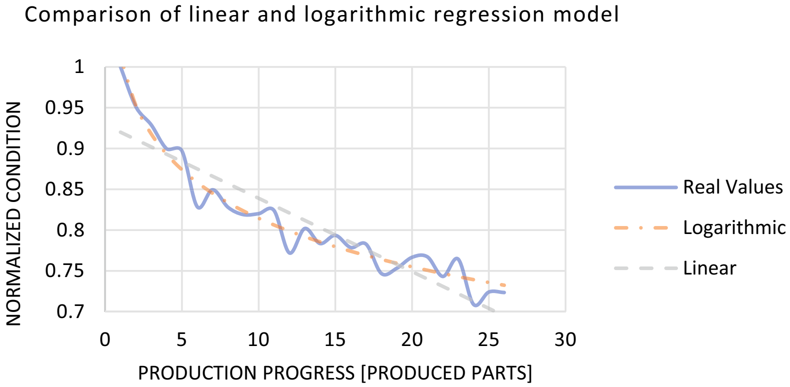


Fig. 3 The derived linear and logarithmic model show a reasonable fit in the training data

Table 1 Comparison of different regression models for the seal degradation behavior

Linear Regression	Non-linear Regression	Multiple regression

RMSE = 26.7
Preparation Effort = Low
Acceptance = Very low
	RMSE = 26.6
	Preparation Effort = Low
	Acceptance = Very low
		RMSE = 3.3
		Preparation Effort = High
		Acceptance: High

show a good fit, but loose accuracy on new data. The RMSE are very low and indicate clear overfitting. From a user's perspective, the acceptance is very low and the generated models will never reach the shopfloor. The multiple regression incorporated approximately 15 variables that allowed a better prediction of the seal condition. The data collection, preparation and modelling effort were significantly higher for the multiple regression, as more factors were included. The cost-benefit ratio was in favor for the data-centric approach, as robustness and applicability are needed to generate benefits from the actual effort.

4 Conclusions & Outlook

Applying machine learning to the RTM process requires integration of disparate systems across the process chain. Capturing domain expert knowledge is the key to tailor data collection and analysis activities. This involves iterative feedback both in system layout and data collection. A high degree of automation is beneficial both to avoid operator variability as well as to formalize the process steps in preparation of data analysis. The high quantity of experiments required to approach industrial settings and relevant machine learning approaches is aided by persistent part identification by embedded bar codes.

Overcoming these challenges allows machine learning to be successfully applied as shown to discover complex interaction of process parameters to enable robust data analysis that is deployable in an industrial setting. More complex approaches such as multiple regression are enabled and show a high acceptance.

Continuing work in this research project will focus on increasing the number of analyses based on the captured data to cover more process features and derive analysis modules with meaningful output to end users.

Supplementing the presented approach future work would benefit from including additional automated quality control methods, adding depth to the captured process data.

Acknowledgements This project has received funding from the Eurostars-2 joint programme with co-funding from the European Union Horizon 2020 research and innovation programme.

The results presented in this paper are based on work done by the authors in cooperation with project partners Hufschmied Zerspanungssysteme GmbH, Bobingen, Germany and Alpex Technologies GmbH, Mils, Austria.

The authors would like to thank Dr. Aaditya Suratkar for proofreading and productive critique of this paper.

References

1. Crawford, B., Rashif, K.M.S., Rashidi, A., Sadiq, R., Milani, A.S.: A Bayesian belief approach to quality control of resin transfer molding process. Int. J. Adv. Manuf. Technol. **109**(7), 1953–1968 (2020). https://doi.org/10.1007/s00170-020-05715-x
2. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data Knowl. Eng. **25**(1), 161–197 (1998). https://doi.org/10.1016/S0169-023X(97)00056-6
3. Kühne, B., Böhmann, T.: Requirements for representing data-driven business models—towards extending the business model canvas. In: Americas Conference on Information Systems 2018: Digital Disruption (2018)
4. Daily, J., Peterson, J.: Predictive maintenance: how big data analysis can improve maintenance. In: Richter, K., Walther, J. (Eds.) Supply Chain Integration Challenges in Commercial Aerospace: A Comprehensive Perspective on the Aviation Value Chain, pp. 267–278. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-46155-7_18.
5. Parida, V., Sjödin, D., Reim, W.: Reviewing literature on digitalization, business model innovation, and sustainable industry: past achievements and future promises. Sustainability **11**(2), 2 (2019). https://doi.org/10.3390/su11020391
6. Chaudhari, R.: Characterization of high-pressure resin transfer molding process variants for manufacturing high-performance composites. Fraunhofer (2014). https://doi.org/10.24406/publica-fhg-280078
7. Fayyad, U.: From data mining to knowledge discovery in databases
8. Plumed, F.M.: CASP-DM: context aware standard process for data mining. Ing. Agua **18**(1), ix (2014). https://doi.org/10.4995/ia.2014.3293
9. Chapman, P., et al.: CRISP-DM: step-by-step data mining guide, p. 76 (1999)
10. Wiemer, H., Drowatzky, L., Ihlenfeldt, S.: Data mining methodology for engineering applications (DMME)—a holistic extension to the CRISP-DM model. Appl. Sci. **9**(12), 2407 (2019). https://doi.org/10.3390/app9122407

Development of a Robotic Bin Picking Approach Based on Reinforcement Learning

Tobias Stuke, Thomas Rauschenbach and Thomas Bartsch

Abstract

Robotic bin picking systems aim to automate the feeding process of randomly stored objects in industrial production. Despite being a research field for decades, there is still a gap between research and industrial application. The presented work intends to improve the utilization of bin picking for the industrial manufacturing of electrotechnical components. In this context, the development process of a system approach based on machine learning is stated. First, related work is presented and the research issue is derived. Second, a comparison between major machine learning techniques with respect to bin picking is made and a reinforcement learning approach is chosen for this work. Therein, a neural network learns strategies for grasping objects from bulk material depending on their position in the bin. Based on manifold states in a simulation environment, it is the goal to gain a versatile character of the robot system. In this regard, preselection criteria, discrete action primitives and grasp constraints are defined that incorporate domain knowledge to shorten the training effort.

T. Stuke (✉)
Technische Universität Ilmenau, Ilmenau, Germany
e-mail: tobias.stuke@tu-ilmenau.de

T. Rauschenbach
Fraunhofer IOSB-AST, Ilmenau, Germany
e-mail: thomas.rauschenbach@iosb-ast.fraunhofer.de

T. Bartsch
Technische Hochschule Ostwestfalen-Lippe, Lemgo, Germany
e-mail: thomas.bartsch@th-owl.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_5

Keywords
Machine Learning • Robotic Bin Picking • Simulation

1 Introduction

Manufacturing companies face increasing challenges in global competition in terms of product variety, demographic change and rising wages [1]. Maintaining competitiveness is often achieved by production automation. Nevertheless, the automated feeding of chaotically stored objects is challenging. As a use case, the manufacturing process of electrotechnical components at Weidmueller Interface GmbH & Co. KG is considered. In this context, specialized solutions for the automated feeding of standardized products in high quantities already exist, such as bowl feeders. However, those solutions cannot be deployed economically for small batch sizes with a high product variety. Consequently, the human eye-hand-coordination is often still required for grasping parts from their random positions and orientations. Even though the automation of bin picking has been researched for decades already [2], the utilization for objects with various shapes in the industrial environment remains a major challenge. This work aims to improve industrial bin picking using a vision-guided robot, by focusing on the decision making process of grasping the objects from the bulk.

2 Related Work

There are two research paradigms of robotic bin picking: 1) rule-based approaches based on analytic metrics to identify suitable grasp candidates, and 2) data-driven approaches that use empirical data to learn the grasping process [3]. In the 1980s, force closure between the object and the gripper was analyzed in order to grasp the object without loosing it. In this context, the ‘wrench’ [4] describes the magnitude of the forces and the torques that can be applied to a grasped object before the transition from static to sliding friction starts. Later, this metric was used to select suitable postures of the gripper. Within the 2010s, data-driven approaches started to gain momentum. Huge datasets have been used in physical experiments, e.g. [5], to learn the ability of grasping objects. There are several reinforcement learning approaches, such as policy-based, value-based or actor-critic algorithms, that have been applied for bin picking. Chen et al. used a Deep Q-Network (DQN) for solving a bin picking task [6]. Tanaka et al. deployed a Proximal Policy Optimization (PPO) for picking and placing items within a logistic process [7]. Furthermore, Ishige et al. utilized a Soft Actor-Critic (SAC) algorithm for picking screws from bulk [8].

2.1 Research Issue

After having identified the current state of bin picking technologies, it is important to understand the requirements for picking randomly stored objects in the industrial environment at Weidmueller. Hence, an expert group from the mechanical engineering department developed a list of criteria. Comparing the criteria pairwise led to the four most important requirements: reliability (18.3%), robustness (17.1%), versatility (14.6%) and domain knowledge (14.6%) [9].

The reliability is calculated as the ratio between the number of successfully picked objects and the number of trials. The robustness of a system characterises the capability to resist varying inputs without changing the initial set of stable parameters. The versatility describes how well the system can adapt to new settings, e.g. grasping unknown objects. The domain knowledge defines information that exceed the scope of the collected sensor data [10], such as the target position for placing the objects.

Next, the four criteria are used to evaluate the related work. The evaluation shows that the strength of the rule-based approaches is domain knowledge, whereas the strength of the learning-based methods is versatility [9]. Thus, both paradigms have different strengths, while no approach meets the requirements entirely. The challenge is to minimize this gap by developing a hybrid approach that combines the individual strengths. Due to the benefit in terms of versatility, a learning-based approach is chosen as the starting point for this work.

2.2 Selection of a Machine Learning Technique

In general, machine learning techniques are classified into supervised, unsupervised and reinforcement learning [11]. Supervised learning (SL) requires labeled output data during training, where the model learns to map input data to the desired output. For instance, the 'Dex-Net 2.0' approach is trained to predict suitable grasp poses based on labeled images [12]. Unsupervised learning (UL), on the other hand, does not require labeled data. The '6-DoF GraspNet' approach, for example, consists of two networks training each other [13]. In contrast, reinforcement learning (RL) relies on an interaction of an agent with an environment. In the Markov decision process (MDP) the agent takes actions and changes the current state of the environment, while a reward signal controls the learning progress in a feedback loop [14].

When comparing the three machine learning techniques, SL and UL are based on visual information solely, while RL relies on the interaction with the environment. Thus, RL has the advantage that the robot kinematic is taken into account. This decreases the solution space, prevents the agent from learning unsuitable grasp poses and thereby shortens the training procedure. On the other hand, SL and UL both provide limited possibilities to include domain knowledge based on images. RL contains design parameters in terms of a reward function and actions, enabling the incorporation of domain knowledge. Consequently, a RL approach is chosen for this work due to the benefits of considering both robot kinematics and domain knowledge.

3 Approach

After the selection of a reinforcement learning approach for grasping randomly stored objects in industrial settings, the approach as well as the procedure and the environment for training the agent are described in the following.

This work considers the subprocesses (I) object detection, (II) grasp planning, (III) path planning and (IV) motion execution. The (I) object detection was developed in a previous master thesis [15] and is applied herein. Based on the detected object poses within the bin, the chosen RL approach performs the (II) grasp planning. The (III) path planning and (IV) motion execution are performed by rule-based algorithms.

3.1 Robotic Bin Picking Based on Reinforcement Learning

The developed bin picking approach is depicted in Fig. 1 and is inspired by the Markov decision process [14]. The input of the agent is the observed state S_t provided by the object detection algorithm [15]. It acquires the six degrees of freedom in terms of the position and orientation of the object, also called pose, in the Euclidean space. Next, a preselection of grasp candidates is executed. Therefore, three criteria were developed in another master thesis [16]: a) centricity, b) height and c) rotation of the object within the bin. The first criteria takes into account that a collision of the robot with the bin is less likely if the object is located in the center of the bin. The second criteria considers the height of the object with respect to the bottom of the bin, since grasping an object occluded by others is difficult. The third criteria evaluates the object according to its rotation. Objects with a high degree of rotation cause difficult configurations of the six-axis robot arm when grasping it. The criteria are multiplied with a weighting coefficient and the product of the three is used for selecting the grasp candidate. Subsequently, the grasp candidate, i.e. the object pose, is fed to the agent. The latter is modeled by a multilayer perceptron (MLP), i.e. a neural network with dense layers solely. It selects an action primitive a_i that is executed in the simulation. Action primitives are elementary trajectories that describe the temporal course of the robots tool center point (TCP) in the action space. A schematic illustration of action primitives is provided in Fig. 2. The action primitives are defined in relation to the object pose. For instance, when executing a 'top down grasp' the approach and remove vector of the TCP is aligned with the object surface normal. Using a set of discrete action primitives allows the incorporation of domain knowledge, since the primitives are designed by the engineer. For performing the selected action primitive in the action space, a rule-based (III) path planning algorithm, such as rapidly-exploring random trees (RRT) [17], is used. It calculates several

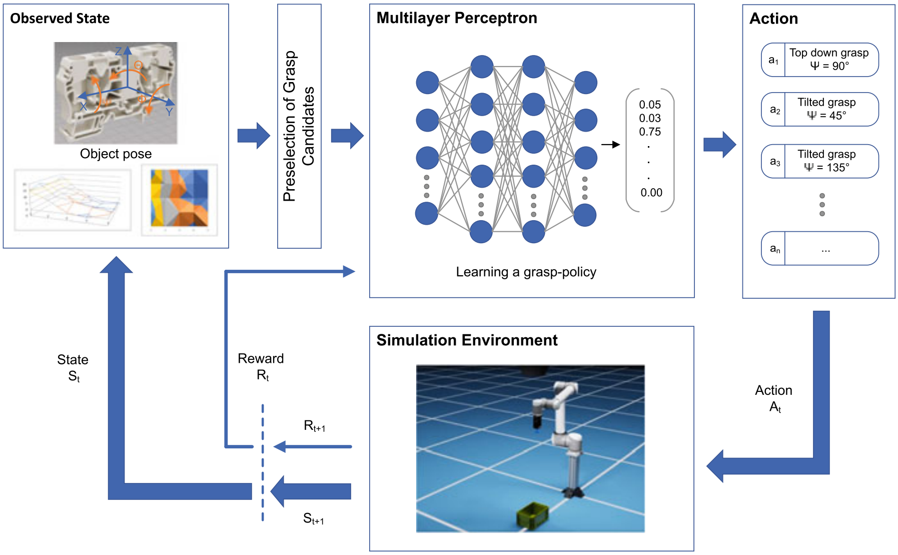


Fig. 1 Robotic Bin picking within the Markov decision process

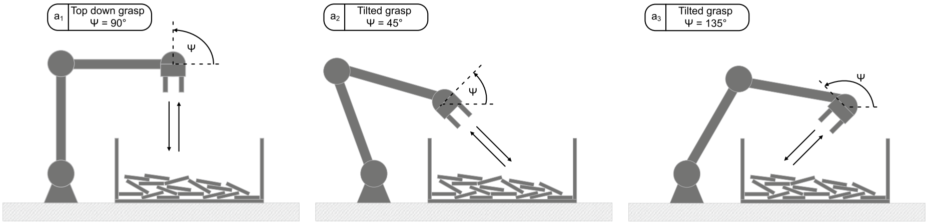


Fig. 2 Schematic Action Primitives for Robotic Bin picking

trajectories that are subsequently evaluated regarding reachability, collision and motion effort. The optimal path is executed (IV) by using a inverse kinematic solver, e.g. [18], that calculates the joint angles of all six robot axes for each time step. The interaction between the robot and the environment leads to a new state S_{t+1} in the Markov decision process (Fig. 1) and the loop repeats with a new observation. Finally, the reward R_{t+1} is fed back to the agent for training the neural network.

3.2 Training Procedure

The goal for the agent is to learn how to choose suitable action primitives a_i from a set A. The updates of the neural network are triggered by the reward signal R_t. In this work, the manipulation task is considered successful if the object is grasped and retracted from the bin. In this context, the sign function [19] provides a traditional method for rewarding the agent. Whether the object was successfully picked from the bin or not, the reward is $R = +1$ or $R = -1$, respectively. Instead of the sign function, an enhanced reward system is developed in this work based on grasp constraints that determine force closure between the gripper and the object. The three constraints are:

$$v_{rel} = 0, \tag{1}$$

$$d_{x-axis} = w_{Object}, \tag{2}$$

$$d_{z-axis} \leq h_{Object}. \tag{3}$$

First, the relative velocity v_{rel} between the gripper and the object needs to be zero. Second, the distance d_{x-axis} between the gripper fingers has to be equal to the object width. Third, the distance d_{z-axis} between the contact point on the gripper finger and the center of the object contact surface needs to be less or equal to the object height. If all three constraints are satisfied, the grasp is considered successful and a positive reward signal $R = +1$ is gained. If one constraint is violated, the agent obtains a negative reward signal $R = -1$. Compared with the traditional reward function calculated at the end of the manipulation task, the three grasp constraints enable a time continuous feedback signal to monitor the force closure grasp during the whole motion execution process. During training, the goal of the agent is to maximize the total sum of rewards by adapting the grasping policy. The policy $\pi\ (s, a)$ describes which action primitive $a_i \in A$ to select, depending on the observed state s, i.e. the object pose in the bin. In the finite MDP, the sets of states S, actions A and rewards R have a finite number of elements. R_t and S_t are random variables of the reward and the state, respectively, at time t (Fig. 1), whereas s and r are particular values of these variables. Given a finite MDP, the value of a state $V(s)$ is calculated according to Eq. 4 as the maximum of the expected reward $\mathbb{E}$ [20].

$$V(s) = \max_{\pi}\ \mathbb{E}\left(r_0 + \sum_{k=1}^{n} \gamma^k\ r_k \mid s_1 = s'\right);\ k = 1, ..., n; \text{with } k, n \in \mathbb{N} \tag{4}$$

Here, r_0 denotes the initial reward and γ refers to a discount rate. The factor $\gamma \in \mathbb{R} \in (0, 1)$ discounts future rewards, taking the economic principle into account that current rewards are more valuable than future rewards [20]. During training, the iterative execution of the bin picking task and the calculation of the argument that maximizes the value function $V\ (s)$ leads to the optimal grasping policy $\pi\ (s, a)$. The parameters of the neural network are updated for each iteration based on the current value $V(s)$ and thereby the agent is trained.

3.3 Training Environment

Since training the agent by interacting with the industrial environment is very expensive, a simulation environment is chosen instead. Compared to training in the industrial environment, there are multiple advantages. First, a collision in the simulation does not cause any damage to the system. Moreover, various states can be generated easily. Thus, training the agent by means of different scenarios aims to generate a behaviour that is robust and generalizes well to the picking task. A simulated environment furthermore enables a parallelization of the training procedure. In this work, the simulation tool 'Isaac Sim' of NVIDIA [21] is used to a) depict the physical properties of the robot, b) implement state of the art RL-algorithms, and c) illustrate visual features with photorealistic representations through its raytracing capability, thereby enabling object detection.

4 Conclusion

In this paper, a method for grasping randomly stored objects in the industrial manufacturing of medium production volumes is presented. The main contribution is a closed-loop control build on a reinforcement learning approach trained in the simulated environment 'Isaac Sim'. Therein, a multilayer perceptron learns an optimal grasping policy $\pi\ (s, a)$ depending on a value function. The goal of the training is maximizing the value $V\ (s)$, i.e. the expected reward r, by adapting the policy $\pi\ (s, a)$. In this work, the reward is correlated to three constraints that determine a force closure grasp. By that, a continuous feedback signal is gained that can be monitored during the whole manipulation process. In addition to the grasp constraints, further domain knowledge is included by the preselection criteria for grasp candidates and the set of discrete action primitives. The presented approach differs from existing RL approaches applied for bin picking, e.g. [6, 22], by the incorporation of domain knowledge to this extent. In conclusion, the presented work constitutes a hybrid approach that combines the strengths of a learning-based approach and a rule-based method in terms of versatility and domain knowledge and by that aiming to minimize the gap between the research and the industrial application of robotic bin picking.

In future work, the developed approach will be trained and tested in the simulation environment. As a test scenario, grasping terminal blocks at Weidmueller will be used. First of all, the focus will be on shaping and testing suitable action primitives. Then, the reward function based on the three grasp constraints will be examined. In this context, the 'wrench' offers a further advancement to monitor force closure during the grasping process. Finally, the deployment of the approach in the industrial production of Weidmueller is planned by using the robot operating system (ROS) as an interface between the simulated and the physical environment.

Acknowledgements This work was developed in collaboration with Weidmueller Interface GmbH & Co. KG.

References

1. Bischoff, J., Hillmann, M., Lübkemann, J., Nyhuis, P.: Fabriken im Wandel. https://www.agiplan.de/wp-content/uploads/2019/05/Fabriken_im_Wandel_2016.pdf. Accessed 22 May 2023
2. Schraft, R., Ledermann, T.: Intelligent picking of chaotically stored objects. Assem. Autom. **23**(1), 38–42 (2003). https://doi.org/10.1108/01445150310460079
3. Sahbani, A., El-Khoury, S., Bidaud, P.: An overview of 3D object grasp synthesis algorithms. Robot. Auton. Syst. (2011). https://doi.org/10.1016/j.robot.2011.07.016
4. Ferrari, C., Canny, J.: Planning optimal grasps. In: Proceedings of the 1992 IEEE International Conference on Robotics and Automation Nice, France, pp. 2290–2295 (1992). https://doi.org/10.1109/ROBOT.1992.219918
5. Levine, S., Pastor, P., Krizhevsky1, A., Ibarz, J., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int. J. Robot. Res. **37**(4–5), 421–436 (2018). https://doi.org/10.1177/0278364917710318. Accessed 22 May 2023
6. Chen, J., Fujinami, T., Li, E.: Deep bin picking with reinforcement learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR. Vol. 80 (2018). https://thisisjeffchen.com/publication/bin-picking.pdf. Accessed 22 May 2023
7. Tanaka, T., Kaneko, T., Sekine, M., Tangkaratt, V., Sugiyama, M.: Simultaneous planning for item picking and placing by deep reinforcement learning. In: 2020 International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, pp. 9705–9711 (2020). https://doi.org/10.1109/IROS45743.2020.9340929
8. Ishige, M., Umedachi, T., Ijiri, Y., Taniguchi, T., Kawahara, Y.: Blind bin picking of small screws. In: 2020 IROS, pp. 9337–9344 (2020). https://doi.org/10.1109/IROS45743.2020.9341423
9. Stuke, T., Bartsch, T., Rauschenbach, T.: Adaptiver Griff-in-die-Kiste – Die methodische Lücke zwischen Forschung und Industrie. In: 18. AALE-Konferenz (2022). https://doi.org/10.33968/2022.14
10. Wirtschaftslexikon, „Kontextwissen". http://www.wirtschaftslexikon.co/d/kontextwissen/kontextwissen.htm. Accessed 3 May 2023
11. VDMA Quick Guide – Machine Learning im Maschinen- und Anlagenbau. VDMA Software und Digitalisierung, p. 6 ff. (2018). https://www.vdma.org/documents/34570/1052572/Quick-GuideMachineLearning-KI.pdf/8021ab42-79a6-5c72-c4f6-20b6c49ad54a. Accessed 2 May 2023
12. Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Aparicio Ojea, J., Goldberg, K.: Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds. CoRR (2017). http://arxiv.org/abs/1703.09312
13. Mousavian, A., Eppner, C., Fox, D.: 6-DOF GraspNet: variational grasp generation for object manipulation. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2901–2910 (2019). https://doi.org/10.1109/ICCV.2019.00299
14. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction, 2nd edn. MIT Press, Cambridge (2020). https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6260249. Accessed 22 May 2023
15. Halle, M.: Entwicklung eines automatisierten Verfahrens zur flexiblen Identifikation und Lokalisierung von Objekten in chaotischer Bereitstellung, TH OWL (2022)

16. Wöltje, F.: Entwicklung eines automatisierten Systems zur Verarbeitung von Objekten in chaotischer Bereitstellung, TH OWL (2022)
17. LaValle, S., Kuffner, J.: Rapidly-exploring random trees. In: Algorithmic and Computational Robotics: New Directions 2000 WAFR (2001). http://msl.cs.uiuc.edu/~lavalle/papers/LavKuf01.pdf. Accessed 22 May 2023
18. NVIDIA: Lula kinematics solver—omniverse robotics documentation. https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/advanced_tutorials/tutorial_motion_generation_lula_kinematics.html. Accessed 22 May 2023
19. Lee, V.: Activation functions—machine learning note documentation. https://machine-learning-note.readthedocs.io/en/latest/basic/activation_functions.html. Accessed 26 May 2023
20. Brunton, S., Kutz, J.: Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, 2nd edn. Cambridge University Press, Cambridge (2022)
21. NVIDIA: NVIDIA Omniverse® Documentation What is Isaac Sim? http://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/overview.html. Accessed 2 Jan 2023
22. Zeng, A., Song, S., Welker, S., Lee, J., Rodriguez, A., Funkhouser, T.: Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In: International Conference on Intelligent Robots and Systems (IROS), Madrid, pp. 4238-4245 (2018). https://doi.org/10.1109/IROS.2018.8593986

Control Reconfiguration of CPS via Online Identification Using Sparse Regression (SINDYc)

Benjamin Kelm, Stephan Myschik and Oliver Niggemann

Abstract

Cyber-physical systems are becoming increasingly complex and prone to faults. To effectively handle these faults, online identification and reconfiguration of the system are crucial. This paper proposes a method for controlling reconfiguration by identifying faults in cyber-physical systems online. The approach utilizes sparse regression (SINDYc) to identify the system dynamics, including faults, and adjusts the control law accordingly by leveraging plant redundancies.

To illustrate the fault handling approach, the study focuses on a well-known control systems example, the inverted pendulum on a cart, which is nonlinear and unstable. By injecting a perturbation signal, the closed-loop system dynamics are separated into input and system dynamics. The SINDYc algorithm is then applied to the measurement vectors of input and output signals, generating an up-to-date dynamic model that incorporates possible faults. In the event of an actuator fault, the identified model is used to reconfigure the control using the Pseudo-Inverse method, optimizing the utilization of available redundancies. Both abrupt and incipient faults in the actuator dynamics are considered in this study.

B. Kelm (✉) · S. Myschik
Institute for Aeronautical Engineering, Universität der Bundeswehr München, Neubiberg, Germany
e-mail: benjamin.kelm@unibw.de

S. Myschik
e-mail: stephan.myschik@unibw.de

O. Niggemann
Fakultät für Maschinenbau, Helmut Schmidt University, Hamburg, Germany
e-mail: niggemao@hsu-hh.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_6

The online identification is limited to linear models in this work, and a full-state feedback controller is reconfigured under the assumption of full observability of the system. A parameter study demonstrates the influence of perturbation signal power and measurement noise on the identifiability of the closed-loop system. Based on the results, it is concluded that the online control reconfiguration approach satisfactorily handles actuator faults in the studied use-case. Furthermore, it can be easily extended to nonlinear model identification and subsequent reconfiguration of nonlinear controllers, such as MPC or INDI.

Keywords

CPS Identification • Sparse Regression • Closed-loop Online Control Reconfiguration

1 Introduction

Cyber Physical Systems (CPS) are prone to various faults caused by failing actuators, sensors, or structural components. Modern systems are becoming increasingly larger and complex, making manual fault handling costly and time-consuming. To enable systems to adapt autonomously to faults, fault-tolerant control (FTC) is necessary. For severe faults, such as the loss of a complete actuator, a new control law is required. This process is called reconfiguration and has the goal of identifying a new valid control law that can restore operation [1].

Due to the immense cost reduction and miniaturization of embedded controllers and sensors in the last decade, data and computational resources are widely available. This allows for computationally expensive algorithms to be incorporated in CPS of arbitrary size and led to the development and implementation of advanced model-based fault-tolerant control architectures, such as explicit model following (EMF), Linear Virtual Actuators (LVA) or Fault-tolerant Model Predictive Control (MPC) approaches [2].

However, all these model-based FTC strategies rely on accurate dynamic models to reconfigure the controller that is handling the faulty system. Generally speaking, the impact of various faults on the system dynamics are not known a-priori. Thus, when developing a comprehensive fault handling strategy, a model of the dynamics of the faulty system needs to be identified online. This requires the algorithm for model identification to be executed online and to rapidly convergence (low data requirement) to allow for a quick reconfiguration. Additionally, a high robustness against noise is desirable to be applicable in practice. In the last years, the well-established research field of system identification has received renewed attention with the rise of data-driven modeling, advanced machine learning algorithms and new findings in the area of compressed sensing [3]. In this paper we strive to present a model-based fault handling strategy for a classical control system, the Inverted Pendulum on a Cart (IPoC) with threefold actuator redundancy. The required dynamic model of the faulty system is identified online through the sparse regression algorithm SINDYc [4]. The identified model of the faulty system is then used for a model-matching reconfiguration algorithm, the Pseudo-Inverse Method [5, p. 391] to reconfigure a full-state feedback (FSF) controller. In this work, we will limit our online identification to linear models, although SINDYc readily extends to nonlinear models. Further, we assume full observability of the system. The continuous plant model, the discrete controller and reconfiguration are implemented in MATLAB/Simulink. In the following we aim to address the following research questions:

1. Are sparse regression algorithms, such as SINDYc, suited to identify models of closed-loop CPS online?
2. What perturbation strategy ensures continuous and adequate identification of the faulty system dynamics?

The paper is structured as follows: After a section on the related work 2, we will first describe the IPoC system followed by a brief overview of the SINDYc algorithm. Section 5 describes the control reconfiguration methodology and architecture. The results will be discussed in Sect. 6 and followed by a discussion and conclusion.

2 Related Work

2.1 Model-Based Fault Tolerant Control

To achieve fault tolerance, technical systems are typically augmented with a Fault Detection, Identification and Recovery (FDIR) scheme. If the fault is severe (e.g. actuator breakdown) and requires a redesign of the control-loop, control reconfiguration is necessary. Otherwise, a control accommodation approach is sufficient by means of robust or adaptive control. Blanke et al. [5] have described multiple approaches to model-based control reconfiguration [5]. Generally, the model-matching design is prevailing in the literature, where the controller of the faulty plant is reconfigured, so that the closed-loop characteristics of the system are consistent to the non-faulty behavior. Beside the Pseudo-Inverse approach presented in this work, the Markov Parameter approach or Linear Virtual Actuators (LVA) approach are alternative model-based reconfiguration methods [5]. In the aerospace sector, a historically strong field of applied FTC, many control architectures and approaches have been developed, as exemplified in [6]. Lombaerts et al., for example, applied a the model reference adaptive control approach to a flight control [7].

Beside the classical FTC approaches originating from control theory, another research field has evolved, mainly dealing with hybrid systems and employing Formal Methods to find a valid configuration [8]. For example, the AutoConf algorithm by Balzereit et al. transfers the reconfiguration problem into a logical formula, which then can be solved using Satifiability Theory (SAT) [9]. Despite the many different approaches to model-based control reconfiguration, the need of an up-to-date system model is ubiquitous, since a system subject to faults is inherently time-variant and faults are not known beforehand.

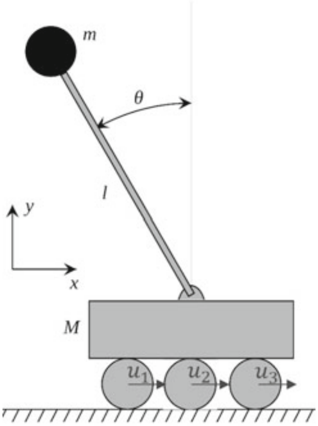
2.2 Online, Closed-Loop System Identification

Closed-loop online system identification has been of interested in many areas. Ljung renders a comprehensive view on system identification and specifically covers the problem of identifiability of closed-loop systems [10, Sect. 13.4]. Leveraging advances in sparse sensing and machine learning, Brunton, Proctor and Kutz have developed the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm [11]. The algorithm employs the thresholded sequencial least-squares regression algorithm, resulting in models that are sparse and thus balance model complexity with descriptive ability. The algorithm then has been further developed to include forced nonlinear systems by extending the function base to include control inputs [4]. The resulting SINDYc formulation has been applied to numerous problems, including a Model Predictive Control (MPC) in the low-data limit by Kaiser, Kutz and Brunton [12]. However, to the best of our knowledge, data-driven sparse regression system identification techniques, such as SINDYc, have not yet been employed explicitly in the context of online closed-loop Control Reconfiguration specifically.

3 System Description and Modeling

Figure 1 shows the redundantly actuated IPoC system, which will constitute the use-case in this study.

Fig. 1 Redundantly actuated Inverted Pendulum on a Cart (IPoC)



Mathematically, the IPoC is described by the following differential equations [3]:

$$\begin{aligned}
\dot{x} &= v \\
\dot{v} &= \frac{-m^2\, l^2\, g\, cos(\theta)\, sin(\theta) + m\, l^2\, (m\, l\, \omega^2\, sin(\theta) - \delta\, v) + m\, l^2\, (1/D)\, F_t}{m\, l^2\, (M + m\, (1 - cos(\theta)^2))} \\
\dot{\theta} &= \omega \\
\dot{\omega} &= \frac{(m + M)\, m\, g\, l\, sin(\theta) - m\, l\, cos(\theta)\, (m\, l\, \omega^2\, sin(\theta) - \delta\, v) - m\, l\, cos(\theta)\, F_t}{m\, l^2\, (M + m\, (1 - cos(\theta)^2))}
\end{aligned} \tag{1}$$

where x is the cart position, v is the velocity, θ is the pendulum angle, ω is the angular velocity. The parameters describe pendulum mass m, the cart mass M, pendulum arm length l, gravitational acceleration g, and the velocity-proportional (Stoke's law) friction damping δ on the cart. The system control input $F_t = \sum_{i=1} F_i$ is compound force applied to the cart and is a linear combination of the three redundant actuators. The actuators are modeled as ideal linear force actuators, not introducing any additional dynamics into the plant. The system is implemented as a nonlinear system in MATLAB/Simulink.

4 Closed-Loop System Identification with SINDYc

Sparse Identification of Nonlinear Dynamics with control (SINDYc) extends the original SINDy algorithm to include external inputs and feedback control [4].

4.1 Sparse Identification—SINDYc

In this section, a brief overview of the sparse regression technique used in SINDYc is presented. Given a dynamical nonlinear system of the form

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2}$$

where $\mathbf{x}(t)$ denotes the state of a system at time t, and the function $\mathbf{f}(\mathbf{x}, \mathbf{u})$ describes the dynamic constraints. The time-dependency is omitted for better readability.

Then, to identify the system from data, we first collect m measurements of the state history $\mathbf{X} \in \mathbb{R}^{m \times n}$, corresponding to n state-vector measurements $\mathbf{x}_i \in \mathbb{R}^m$ and the input history $\boldsymbol{\Upsilon} \in \mathbb{R}^{l \times m}$.

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ | & | & & | \end{bmatrix}, \boldsymbol{\Upsilon} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_l \\ | & | & & | \end{bmatrix}, \tag{3}$$

These measurement matrices can be expanded with nonlinear candidate functions f into a library matrix $\mathbf{\Theta}(\mathbf{X}, \mathbf{\Upsilon})$,

$$\mathbf{\Theta}(\mathbf{X}, \mathbf{\Upsilon}) = \begin{bmatrix} | & | & | & & | & \\ \mathbf{1} & \mathbf{X} & \mathbf{\Upsilon} & \dots & f(\mathbf{X}, \mathbf{\Upsilon}) & \dots \\ | & | & | & & | & \end{bmatrix} \tag{4}$$

which in turn constitutes the function basis to regress upon, so that the regression problem is defined as

$$\dot{\mathbf{X}} = \mathbf{4}\mathbf{\Theta}^T(\mathbf{X}) \tag{5}$$

where Ξ denotes the sparse coefficient matrix to be determined, and $\dot{\mathbf{X}}$ describes derivative of the state history matrix with respect to time, which can be measured or computed. The resulting over-determined system can be solved efficiently with the thresholded sequential least-squares algorithm, a relaxation of the Sparse Regularized Regression Problem, as described by Zheng et al. [13].

4.2 Identifiability in Closed-Loop Systems

As shown by Ljung, for a well-defined closed-loop system that exhibits integrative behavior (contains a delay) and is stable, the convergence theorem generally holds under the additional assumption that the data is informative and the model set contains the true system [10, p. 430].

Therefore it is possible to identify a system under closed-loop feedback control from data. However, it is more challenging to obtain informative data, since an important purpose of feedback is to make the closed-loop system less sensitive to disturbances.

As indicated by Brunton et al. SINDYc can be applied to closed-loop systems, but requires an additional input perturbation [4]. The key issue is that for closed-loop systems, where $\mathbf{u} = \mathbf{K}(\mathbf{x})$, with $\mathbf{K}$ being the feedback gain matrix, the regression problem becomes ill-conditioned and the coefficients matrix Ξ thus rank-deficient. This is because it is impossible to disambiguate the influence of the feedback control and the internal dynamics. The single requirement for obtaining informative data from closed-loop systems is that the input must be be persistently exciting of a certain order, i.e. that it contains sufficiently many distinct frequencies [10, p. 415]. In the case of online system identification, the perturbation strategy is a tradeoff between optimal identifiability (high input power) vs. optimal reference tracking (smallest possible input power).

5 Control Reconfiguration

The control reconfiguration architecture is detailed in Fig. 2. Generally, a dynamic system can experience faults in three locations: actuator, plant, and sensor faults. Only actuator faults

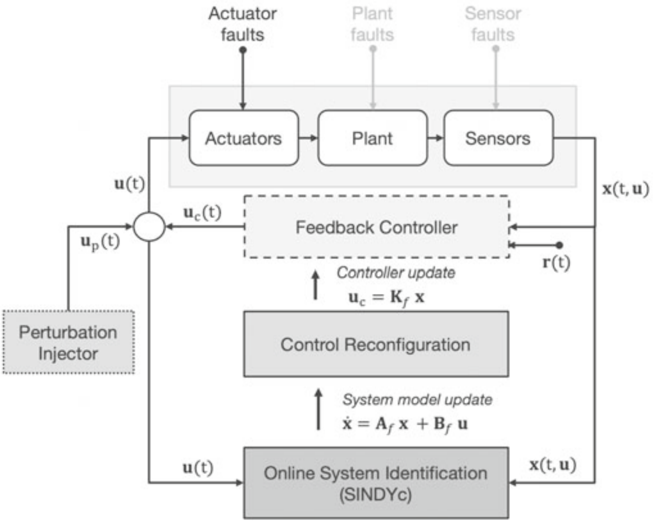


Fig. 2 Online Control Reconfiguration architecture for a linear full state feedback controller

are considered in this work. To reconfigure the controller in the case of a fault, the online identification layer first estimates a new system model of the faulty system $\dot{\mathbf{x}} = \mathbf{A}_f\mathbf{x} + \mathbf{B}_f\mathbf{u}$ based on recent system behavior, described by the input $\mathbf{u}(t)$ and states $\mathbf{x}(t, \mathbf{u}, f)$ time series. To disambiguate between the full state feedback controller dynamics and the plant dynamics the perturbation injector excites the system directly and continuously. The new dynamic model is then served to the control reconfiguration layer. Depending on the severity of the fault, the reconfiguration algorithm reallocates resources by altering the full state feedback controller matrix K_f.

For simplicity, a model-matching reconfiguration strategy is chosen, based on the Pseudo-Inverse method detailed in [5, pp. 391–394]. Given a linear state-space model of a nonlinear dynamic system unsubject and subject (index f) to a fault

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t), \quad \dot{\mathbf{x}}(t) = \mathbf{A}_f\,\mathbf{x}(t) + \mathbf{B}_f\,\mathbf{u}(t) \tag{6}$$

a state feedback controller for the faulty plant can be readily expressed. The optimal gains are determined by solving the Riccati Equation for a Linear Quadratic Regulator (LQR) [3]

$$\mathbf{u}(t) = -\mathbf{K}_f\,\mathbf{x}(t), \tag{7}$$

where $\mathbf{K}_f$ denotes the control matrix for the faulty plant. To achieve similar closed-loop dynamics, the difference between the two system models needs to be minimized. The solution to the unknown control matrix $\mathbf{K}_f$ is then given by:

$$\mathbf{K}_f = \mathbf{B}_f^{+}(\mathbf{A}_f - \mathbf{A} + \mathbf{B}\,\mathbf{K})f \tag{8}$$

with the Pseudo-Inverse $\mathbf{B}_f^+$ of the input matrix. The resulting control matrix minimizes the differences of dynamical properties between the nominal loop and the faulty loop and gives the best possible controller design for the linear system subject to a fault.

6 Results

This section presents first results of a parameter study for the identification of the IPoC system with SINDYc, yet without reconfiguration. The second section then presents the results of the complete reconfiguration of IPoC system.

6.1 Closed-Loop Identification Parameter Study

There are a variety of parameters that are relevant to the identification process of closed-loop systems. In this section we will study effect of the perturbation signal power and the signal-to-noise ratio (SNR) for the measured sensor signals used for identification. There exists a number of different signal types like White Gaussian Noise (WGN), Pseudo-Random Binary Sequence (PRBS) or multisine signals.

However, we found that, for our purpose, the perturbation signal type, after being bandpass filtered, has only marginal influence on the identification results. To reduce the parameter space we will therefore only consider the bandpass-filtered WGN signal. To ensure comparability for different frequency bands, the power of the perturbation signal $P(u_p)$ is normalized. Figure 3 shows the unfiltered and filtered WGN perturbation signal and a frequency band of $f_b = 2..15$ Hz.

For the parameter study the perturbation signal power was varied from $P(u_p) = 0.01..0.63$ W $= 10..28$ dBm and a signal-to-noise ratio from $SNR_{id} = 100$ to 10 was studied.

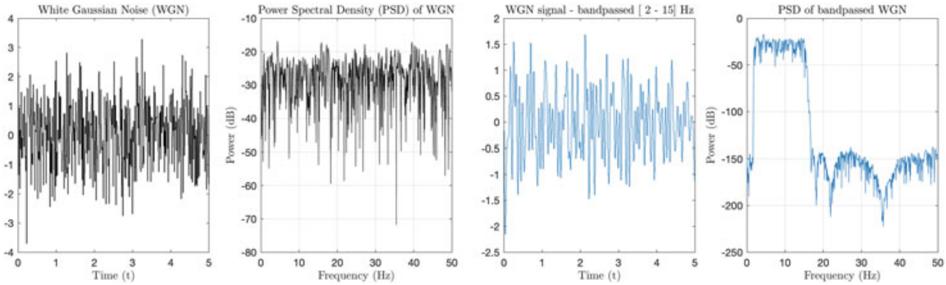


Fig. 3 Filtered White Gaussian Noise (WGN) as perturbation signal

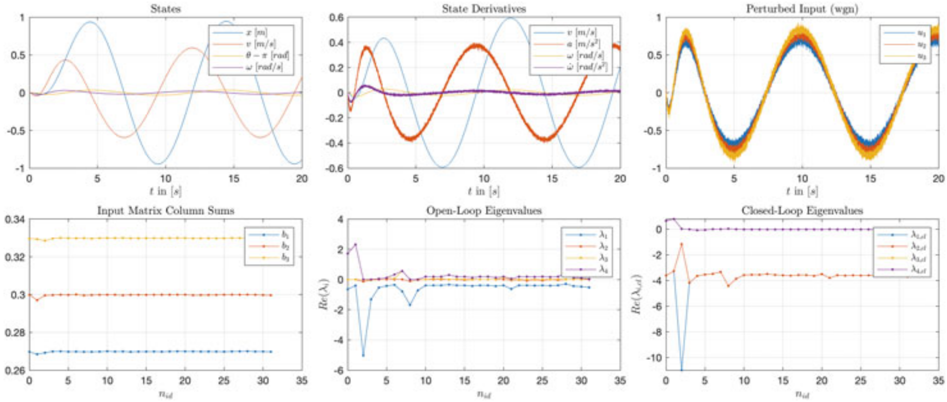


Fig. 4 Parameter study case for IPoC system identification

For all experiments, a sinusoidal reference signal with frequency $f_r = 0.1$Hz is imposed on the system.

Figure 4 shows the simulation and the resulting identification plots for a signal power of $P(u_p) = 14.5$ dBm and a SNR of 100. The states, derivatives and input plots on the top show the closed-loop system response to the reference signal. On the bottom left, the identified column sum b_i of the input matrix (pertaining to one of the three actuators each) is plotted for progressive identification timeframes n_{id} corresponding to $N_t f = 300$ samples at a sampling frequency of $f_s = 500$ Hz each. The identified open loop eigenvalues λ_i are plotted in the bottom center plot, and the closed-loop eigenvalues $\lambda_{i,cl}$ on the right.

The input matrix corresponding to the actuator effectiveness is identified correctly throughout the experiment and exhibits low variance. The system matrix is not identified correctly and exhibits high variance for the first few identification timeframes. This is probably due to settling effects, i.e. that the systems initial conditions are equal to the "pendulum-up" fixpoint, but the reference signal demands a constant velocity initially, resulting in large gradients. For later timeframes, however, the system is identified correctly.

The semilog-plot in Fig. 5 shows the statistical evaluation of the identification results for varying perturbation signal power and Signal-to-Noise ratios for a perturbation frequency band of $f_b = 2..15$ Hz. The different signal power levels are shown by the different colors. The shaded areas correspond to the variance and the solid line to the mean of all $n_{id} = 32$ identification timeframes of one parameter set.

The left plot shows the identification results of the only the first actuator to reduce clutter. For low perturbation signal powers, a significant increase in variance can be seen for a SNR of 30 and below. For higher signal power the identification remains at low variance even for very low SNR. It is important to note, that the algorithm overestimates the effectiveness consistently. A similar behavior is observed for the open-loop and closed-loop identification of the system matrix. For low perturbation signal powers, the eigenvalues are

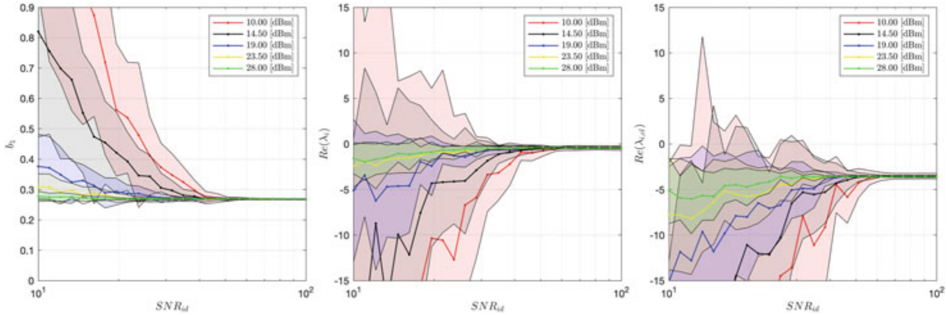


Fig. 5 Statistical evaluation of identification quality for varying perturbation signal power and Signal-to-Noise ratio (SNR) for a perturbation frequency band of $f_b = 2..15$ Hz

poorly identified for a SNR of 40 and below. For higher perturbation signal powers, the eigenvalues are identified with less variance. Here, a clear tendency towards overestimating the system dynamics is observed both for lower SNR as well as lower perturbation signal powers.

Further experiments have shown that there is an optimal frequency band of the perturbation signal. Both very low and very high frequencies lead to identification results with significantly higher variance.

6.2 Closed-Loop Identification and Control Reconfiguration

In this section we will simulate the complete the identification and reconfiguration strategy for the closed-loop IPoC system subject to both abrupt and incipient actuator faults.

Figure 6 shows the simulated system response of the the closed loop system. The top three graphs show the states, the derivatives and the forces on the cart, which are produced by the actuators subject to faults. At time $t = 5$ s an abrupt fault is introduced to actuator 3, and at time $t = 10$ s an incipient fault is introduced to actuator 2, linearly loosing effectiveness over $\Delta t = 10$ s.

The bottom three graphs show the online identification and reconfiguration results. The input matrix is identified, as expected, with low variance. Both the abrupt and incipient actuator faults are caputured well. The system matrix (bottom center graph) is identified with high variance, and the corresponding eigenvalues only settle after the faults have transpired. The bottom left graph shows the row sums of the reconfigured full state feedback controller matrix $\mathbf{K}_f$. The reconfiguration of the relevant entries is clearly seen, as the controller is adjusted to changing plant behavior. Despite the full failure of two actuators, the closed-loop system remains stable and retains it's pre-fault characteristics.

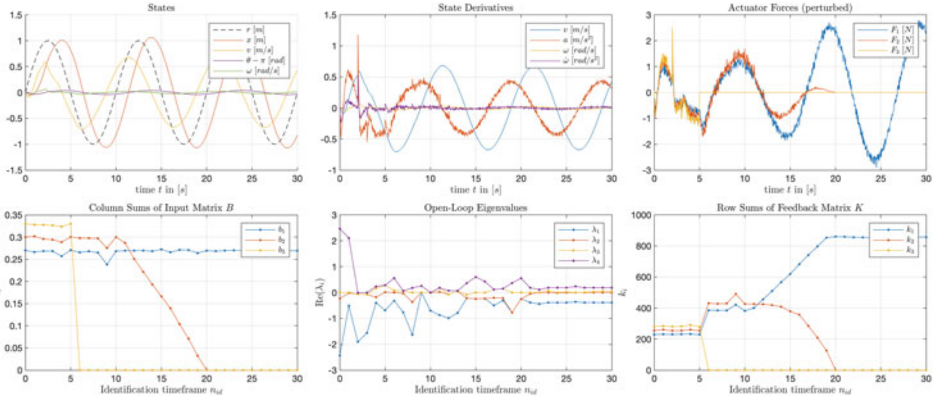


Fig. 6 Perturbed IPoC simulation data with fault injection and reconfiguration, $P(u_p) = 20\,\text{dBm}$, $SNR = 80$

7 Limitations and Outlook

The presented approach for fault-handling by control reconfiguration via online identification with SINDYc has shown general applicability to closed-loop cyber-physical systems, such as the Inverted Pendulum on a Cart. In order to successfully identify a closed-loop system, the perturbation signal needs to be chosen carefully. It has been shown that the higher the perturbation power and signal-to-noise ratio, the better the performance of SINDYc. What remains unclear why both the input matrix coefficients and the stability of the system (more negative eigenvalues eigenvalues) are overestimated consistently. Also, what has not been shown sufficiently is the dependency upon the perturbation frequency-band. Failures in plant dynamics (e.g. changing pendulum mass) have also not been considered. However, the presented fault handling strategy readily integrates these types of faults as well.

The presented approach is limited in several ways: First of all, the requirement of full observability strongly limits the applicability for many real-world systems. On this end, a combination with a estimator-filters (e.g. Kalman filter) would relax this requirement. This would also allow for treatment of sensor faults. Second, the requirement of consistent excitation compromises tracking capabilities of the system and poses increased strain on the actuators. Future implementations could make use of external anomaly detection algorithms to trigger a system “self-test” consisting of a perturbation, identification and possible reconfiguration. Thirdly, the limited identifiability of the system matrix **A** restricts the application for weakly actuated systems.

Future research could focus on the implementation of this approach on a real-world system with real, noisy data. The influence of actuator dynamics or explicit time-delays on the identifiabiliy might be worthwile to study. Further, the identification of the optimal perturbation frequency-band, or alternative perturbation strategies, like a pulsed and synchronized perturbation and identification approach, would present an interesting research opportunity. Finally, a combination of the presented approach with input linearization techniques, such as Incremental Nonlinear Dynamic Inversion (INDI), where only the input matrix is required, offers a very promising nonlinear fault-tolerant control strategy.

Acknowledgements This research is funded by (K)ISS as part of dtec.bw®—Digitization and Technology Research Center of the Federal Armed Forces of Germany (BMVg) which we gratefully acknowledge [14].

References

1. Blanke, M., Christian Frei, W., Kraus, F., Ron Patton, J., Staroswiecki, M.: What is fault-tolerant control? IFAC Proc. Vol. **33**(11), 41–52 (2000). https://doi.org/10.1016/S1474-6670(17)37338-X
2. Eltrabyly, A., Ichalal, D., Mammar, S.: Fault-tolerant model predictive control trajectory tracking for a quadcopter with 4 faulty actuators. IFAC-PapersOnLine **54**(4), 141–146 (2021). https://doi.org/10.1016/j.ifacol.2021.10.024
3. Brunton, S.L., Kutz, J.N.: Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge University Press, Cambridge (2019)
4. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Sparse identification of nonlinear dynamics with control (SINDYc). IFAC-PapersOnLine **49**(18), 710–715 (2016). https://doi.org/10.1016/j.ifacol.2016.10.249
5. Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M.: Diagnosis and Fault-Tolerant Control. Springer, Berlin (2016)
6. Edwards, C., Lombaerts, T., Smaili, H., Morari, M., Thoma, M. (eds.): Fault Tolerant Flight Control: A Benchmark Challenge, Lecture Notes in Control and Information Sciences, Vol. 399. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-11690-2, http://link.springer.com/10.1007/978-3-642-11690-2
7. Lombaerts, T., Chu, P., Mulder, J.A.B.: Flight control reconfiguration based on online physical model identification and nonlinear dynamic inversion. In: Morari, M., Thoma, M., Edwards, C., Lombaerts, T., Smaili, H. (eds.) Fault Tolerant Flight Control, Lecture Notes in Control and Information Sciences, Vol. 399, pp. 363–397. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-11690-2_13, http://link.springer.com/10.1007/978-3-642-11690-2_13
8. Campos, J., Seatzu, C., Xie, X. (eds.): Formal Methods in Manufacturing, Industrial Information Technology Series. CRC Press, Boca Raton (2014)
9. Balzereit, K., Niggemann, O.: Autoconf: a new algorithm for reconfiguration of cyber-physical production systems. IEEE Trans. Ind. Inform. 1–1 (2022). https://doi.org/10.1109/TII.2022.3146940
10. Ljung, L.: System Identification: Theory for the User, 2nd ed. Prentice Hall Information and System Sciences Series. Prentice Hall PTR, Upper Saddle River (1999)
11. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016). https://doi.org/10.1073/pnas.1517384113
12. Kaiser, E., Kutz, J.N., Brunton, S.L.: Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proc. R. Soc. A Math. Phys. Eng. Sci. **474**(2219), 20180335 (2018). https://doi.org/10.1098/rspa.2018.0335

13. Zheng, P., Askham, T., Brunton, S.L., Kutz, J.N., Aravkin, A.Y.: A unified framework for sparse relaxed regularized regression: SR3. IEEE Access **7**, 1404–1423 (2019). https://doi.org/10.1109/ACCESS.2018.2886528
14. DTEC – (K)ISS – Künstliche Intelligenz für die Diagnose der ISS (2022). https://dtecbw.de/home/forschung/hsu/projekt-kiss

Using Forest Structures for Passive Automata Learning

Arne Krumnow, Swantje Plambeck and Goerschwin Fey

Abstract

We present an extension on a passive learning algorithm for deterministic finite automata (DFAs) and Mealy machines which is based on the regular positive and negative inference (RPNI) algorithm. The extension builds a structure which is inspired by random forests. Instead of building a single automaton, the forest builds several. The paper introduces two forest structures for learning DFAs, together with their respective extension for Mealy machines, where the choice of which to pick depends on the output to achieve, as well as the type of application. Following that, the empirical analysis shows which parameters yield better results than the basic RPNI passive learning approach. In our experiments, forest structures for passive learning of automata yield significant improvement over the standard RPNI algorithm of up to 43% in the total number of correct outputs when testing Mealy machines.

Keywords

Passive Automata Learning • Forests • RPNI

A. Krumnow (✉) · S. Plambeck · G. Fey
Hamburg University of Technology, Hamburg, Germany
e-mail: arne.krumnow@tuhh.de

S. Plambeck
e-mail: swantje.plambeck@tuhh.de

G. Fey
e-mail: goerschwin.fey@tuhh.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_7

1 Introduction

Modern Cyber-Physical Systems (CPS) often have a high complexity, because they consist of multiple subcomponents including, e.g., digital embedded systems, sensors, and software systems. Additionally, CPS contain interfaces to their physical environment making their functionality dependent on complex physical behavior. These characteristics make modeling of CPS very difficult. Still, models of CPS are needed, e.g., for test, verification, or anomaly detection. Machine learning helps in modeling complex CPS by automatically learning a model of a system from data. This does not only allow to generate knowledge of the system, but also speeds up analysis and design of systems where models are often derived in a time-consuming, manual process. One approach for model learning is automata learning which derives an automaton model from observations of a system. Automata learning has a strong theoretical base coming from the field of language inference, discussed, e.g., in the seminal work of [6]. Techniques, implementing automata learning differentiate between active [3, 12, 18] and passive [11, 14] automata learning. Active learning adaptively updates a model hypothesis by asking the system for new information. This requires to directly perform actions on the original system. Passive automata learning is mostly used if a direct interaction with the system is not possible. Instead, an automaton is learned on a pre-collected learning set. The quality of learned models in this scenario is limited by the amount of data provided for learning. There are several approaches in improving passive automata learning, e.g., to speed-up execution [1] or to handle specific learning data [7, 10, 13].

In contrast to these approaches, our strategy aims at improving the accuracy of learned automaton models and—as will be discussed within the paper—especially support model accuracy in case of limited learning data sets. The work of Gold [6] proves that the established algorithms for passive automata learning yield exact, deterministic automaton models, on a *characteristic learning set*. This means that the training data represents the whole target automaton. For a learning data set collected during operation of the system, getting a characteristic learning set cannot be guaranteed in general.

Our approach combines multiple learned models to improve the overall accuracy. This approach is known in machine learning, e.g., in random forests, combining multiple learned decision trees, first discussed in [4]. Since then, random forests have already been used in multiple applications, e.g., in image processing [5], navigation [2], or energy systems [17]. The performance of random forests illustrates that machine learning methods benefit from randomized learning algorithms when using multiple learned models. An improved accuracy of learned models is important, e.g., if the model is used for monitoring where false alarms or undetected errors should be avoided.

We propose *automata forests* applying this idea to passive automata learning. The main contribution of our paper is a new concept of passive automata learning. This concept combines the internal randomization of the automata learning algorithms and the idea of random forests to consider multiple learned automata for modeling. We build automata forests on top of the Regular Positive and Negative Inference (RPNI) algorithm which is one of the most commonly used passive learning algorithm. We present two different strategies to create automata forests. Furthermore, both of these approaches are applied to learn Deterministic Finite Automata (DFA) and Mealy machines, respectively. Our results imply that this concept is especially helpful when a characteristic learning set is not available. In an experimental evaluation, we observe up to 43% better accuracy of the forest models compared to automata learned with the ordinary RPNI algorithm when applied for prediction on an evaluation set. Section 2 provides some notation and basic information on automata and automata learning. In Sect. 3 the two forest algorithms and their implementation for DFAs and Mealy machines are described. Section 4 gives insight into the performance of the forest algorithms in an experimental evaluation. Finally, Sect. 5 summarizes and discusses the results identifying a superior performance of automata forests compared to the common RPNI algorithm.

2 Preliminaries

Our approach learns DFAs and Mealy machines. A DFA M represents a language $\mathcal{L}$ in the set of regular languages. The automaton decides if a given input word belongs to the language of the automaton M. A Mealy machine $\mathcal{A}$ translates an input word to an output word.

There exist many approaches for learning automata. For this paper, we use the RPNI algorithm [8]. The algorithm uses positive and negative samples, where positive words are element of $\mathcal{L}$ from DFA M and negative words do not belong to $\mathcal{L}$. For DFAs, the learner uses positive and negative sampled words as inputs to construct a tree spanning all given positive samples. Afterwards, the algorithm tries to merge states in the tree, while continuously checking if the merging operation is valid using the negative samples. The validation check is done by looking up if the tree would still reject all given negative input samples after a merge.

For Mealy machines, the learner gets tuples of input and output words sampled from the system to be learned and builds a tree again. Merging operations are now valid if, after a merging step, the output words of the Mealy machine $\mathcal{A}$ given the sampled input words are still equivalent to all corresponding sampled output words.

3 Algorithms for Learning of Automata Forests

3.1 Forest Structure

The general idea behind the algorithms is that multiple automata are built forming the forest structure. Thus, a forest structure consists of several *instances* of automaton learners. Here, each learner implements the RPNI algorithm. Each of the instances inside the forest gets assigned a random fraction of the learning dataset. Afterwards, every instance of the forest builds an automaton. The automata corresponding to each instance are evaluated to form an output. The parameters to design the automata forests in this paper are:

- $m \in \mathbb{N}$ is the number of instances inside the forest,
- $\alpha \in (0, 1)$ is the percentage of the original dataset given to each instance.

In the following, we show two different approaches when making a decision on how to evaluate these forest structures to form an output.

3.2 Forest with Cross Validation (ForestCV)

ForestCV is inspired by the cross-validation method [16]. Every instance of the forest stores the corresponding set from the original learning set, which was not used to train the automaton, as its test set (with the information whether a word is accepted by the target automaton or not). Afterwards, each DFA inside the forest gets scored using the corresponding test set. The scoring gives an automaton a point for every correctly validated word (accept words from the positive sample and decline negative words). ForestCV outputs the DFA, which achieved the highest score using this cross-validation method.

When dealing with Mealy machines, the output of an automaton changes to translating an input to an output word. The basic principle to form a ForestCV for Mealy machines is the same as for DFAs, where the evaluation is covered by the cross-validation and the output Mealy machine is the automaton inside the forest, which performed best given a chosen metric. We analyze the following metrics when translating words: (1) outputting the Mealy machine, which most often correctly translated output words (ForestCV$_W$); (2) outputting the Mealy machine, which had on average the lowest Hamming distance (counting character substitutions when comparing words) to the sampled output words (ForestCV$_{ED}$). The decision which metric to take when constructing ForestCV depends on what the user wants to achieve, i.e., which metric is to be minimized or maximized.

3.3 Forest with Majority Voting (ForestMV)

ForestMV does not output a DFA, but input and output of the forest are equal to a DFA, i.e., the forest decides whether a word is element of $\mathcal{L}$. For an input word, all DFAs decide if they accept or decline the word. The answer which the majority of DFAs take, is the output of the forest. For Mealy machines, given an input word and the output alphabet of the target automaton, ForestMV triggers all Mealy machines inside and forms an output word. This is done by iterating characterwise over the input word. On every character a majority vote picks the output character most often choosen by the Mealy machines, as output character of the forest at the given input character position.

4 Experimental Evaluation

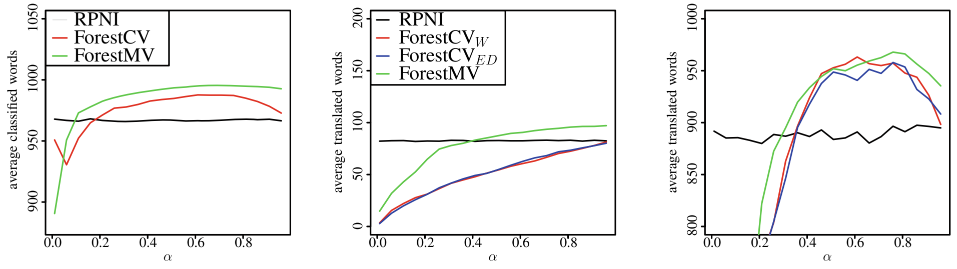
Our empirical evaluation includes two scopes: (1) learning regular expressions. Since regular expressions are equivalent to DFAs, we sample positive words from a regular expression instead of a real target automaton. Negative samples are randomly generated words that do not belong to the considered language $\mathcal{L}$. The 2 regular languages examined in our paper are: $\mathcal{L}_1$ with an alphabet size of 36 and 12074 words, $\mathcal{L}_2$ with an alphabet size of 26 and 7938 words; (2) learning Mealy automata given random target Mealy machines. Many CPS can be abstracted to Mealy machines, identifying discrete in- and outputs. In the experiment, we randomly generate input words on the Mealy machine's alphabet and feed the input words to the target Mealy machine, resulting in sampled output words. The 3 Mealy machines in our paper are: $\mathcal{A}_1$ with an input alphabet of size 10, an output alphabet of size 10, and 20 states. $\mathcal{A}_2$ with an input alphabet of size 10, an output alphabet of size 10, and 5 states. $\mathcal{A}_3$ with an input alphabet of size 15, an output alphabet of size 8, and 15 states.

The words obtained from the target language (1) and Mealy machine (2) form training and test set, respectively to evaluate the different forests. The nominal RPNI algorithm is used as the reference in the evaluation. When building the forests, each instance gets the same amount of words for training. ForestCV and ForestMV use the fraction $\alpha \in (0, 1)$ of the input data to learn their inner m automata. RPNI uses 100% of the input data. Training data used in Sect. 4 are 1000 positive words for $\mathcal{L}_1$, 80 for $\mathcal{L}_2$ with each 500 negative words and 500 pairs of input and output words for $\mathcal{A}_{1,2,3}$. The experiments as well as the implementation of the algorithms are based on the LearnLib framework [9].

4.1 Hyperparameter Tuning

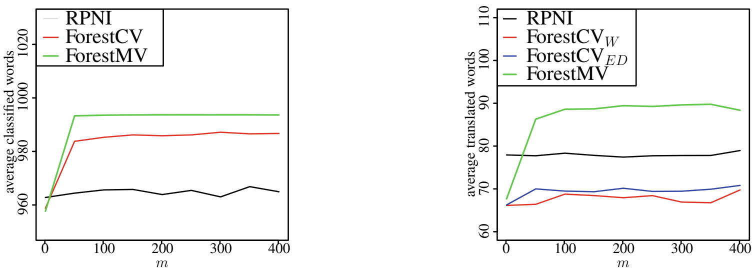
Each test set contains 1000 words and gets continuously tested on the algorithms, while increasing α. The y-axis shows the average number of successes of the algorithms (classifying words in $\mathcal{L}_1$, translating words for $\mathcal{A}_1$, $\mathcal{A}_2$) and the x-axis depicts α. The plots in Fig. 1 show different results on how to choose α. Figure 1a and 1c show that, when steadily increasing α, the curve has a global maximum at roughly $\alpha = 60\%$ for DFAs and $\alpha = 80\%$ for Mealy machines. However, in the case of $\mathcal{A}_1$, the curve increases until $\alpha = 100\%$ (Fig. 1b). For the tests in our paper, we chose $\alpha = 60\%$ for DFAs and $\alpha = 80\%$ for Mealy machines.

Figure 2 shows an analysis of the parameter m. Each test set contains 1000 words and gets continuously tested on the algorithms while increasing m. The y-axis shows the average success rate of the algorithms (classifying words in $\mathcal{L}_1$, translating words for $\mathcal{A}_1$) and the x-axis depicts m. In our experiments the number of successes converges with increasing m.



(a) Increasing α on $\mathcal{L}_1$ (b) Increasing α on $\mathcal{A}_1$ (c) Increasing α on $\mathcal{A}_2$

Fig. 1 Performance analysis with respect to α



(a) Increasing m for $\mathcal{L}_1$ (b) Increasing m for $\mathcal{A}_1$

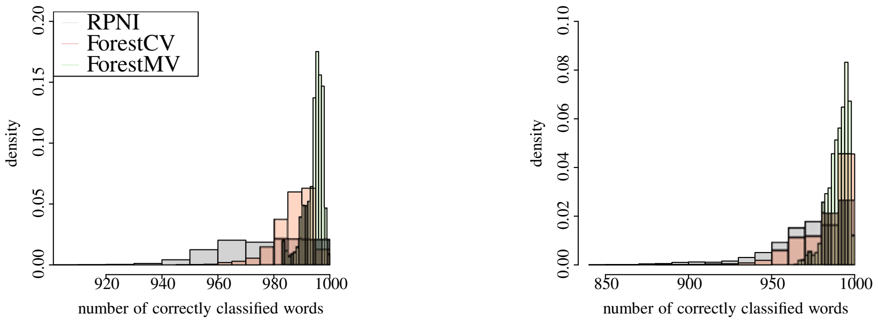
Fig. 2 Performance analysis with respect to m

Figure 2a and 2b show, that after increasing m above 100, the result is not changing much. Because of this, $m = 100$ is chosen in our experiments.

4.2 Analyzing DFAs

In the following, the algorithms are executed with $\mathcal{L}_1$: 5 different training sets which each get 5 different test sets on 100 iterations resulting in 2500 observations and $\mathcal{L}_2$: 10 different training sets on each 10 different test sets with 100 iterations resulting in 10000 observations. Figure 3 shows the results of the DFA examples. The x-axis depicts the number of correctly classified words with a maximum of 1000 and the y-axis depicts the density distribution of the data using the histogram function from R [15], i.e., depicting the probability that a data point achieves a certain number of correctly classified words. In our experiments, the plots show that both forest algorithms outperform the RPNI.

Table 1 analyzes the metrics of the plots from Fig. 3: average (avg), variance (var), false positives (false pos.), false negatives (false neg.) and the execution time in seconds (exec-



(a) $\mathcal{L}_1$ (b) $\mathcal{L}_2$

Fig. 3 Number of correctly associated words for DFA's on 1000 tries

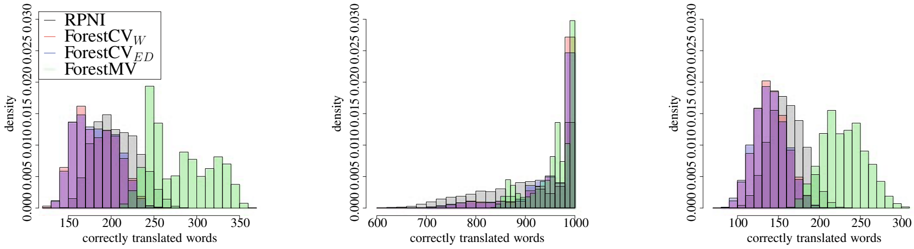
Table 1 Metrics for DFAs

Figure	\	RPNI	ForestCV	ForestMV
Fig 1	avg	975.1	987.4 (+1.3%)	994.8 (+2.0%)
	var	233.0	47.1 (−80.0%)	11.5 (−95.0%)
	false pos.	8e-4	0 (−100%)	0 (−100%)
	false neg.	24.9	12.6(−49.3%)	5.2 (−79.1%)
	execTime	0.0365	2.73 (+7386%)	2.74(+7415%)
Fig 2	avg	972.0	982.9 (+1.1%)	990.4 (+1.9%)
	var	581.7	259.2 (−55.4%)	40.0 (−93.1%)
	false pos.	10.3	6.1 (−41.2%)	2.9 (−72.0%)
	false neg.	17.7	11.0 (−37.9%)	6.7 (−62.3%)
	execTime	0.00459	0.3317 (+7125%)	0.3330 (+7154)

Time) of each algorithm and comparing them to RPNI. The percentage in parantheses shows the approximate increase or decrease of the forests in the metrics (green cell markings are the highest improvement; red cell markings depict the slowest execution time). In the presented results, forest structures show better results with respect to all presented metrics in Table 1, while their execution time is increasing by a factor of roughly ≈ 72.5 compared to the basic RPNI. Experiments in Sect. 4.2 are executed on an Intel© Core™i5-7300HQ CPU @ 2.50GHz processor with 8GB RAM.

4.3 Analyzing Mealy Machines

For learning Mealy machines the test cases are produced by: building $\mathcal{A}_1$ at random 5 different times with each 3 different training sets on each 3 different test sets executing them 100 times resulting in 4500 observations; building $\mathcal{A}_2$ at random 10 times with each 3 different training sets on each 3 different test sets and is executed 100 times at each step which results in 9000 observations; building $\mathcal{A}_3$ at random 25 times and sampled together with each 2 different training sets and each 2 different test sets with 100 iterations for a total of 10000 observations. The x-axis depicts the number of correctly translated words with a



(a) $\mathcal{A}_1$ (b) $\mathcal{A}_2$ (c) $\mathcal{A}_3$

Fig. 4 Number of correctly translated words for Mealy machines on 1000 tries

Table 2 Metrics for Mealy machines

Mealy machine	\	RPNI	ForestCV$_W$	ForestCV$_{ED}$	ForestMV
$\mathcal{A}_1$	avg	206.4	182.6 (−11.5%)	183.2 (−11.2%)	280.1 (+35.7%)
	var	636.2	595.2 (−6.5%)	568.2 (−10.68%)	1272 (+99.9%)
	execTime	0.045	3.31 (+7254%)	3.33 (+7306%)	3.38 (+7413%)
$\mathcal{A}_2$	avg	894.4	954.3 (+6.7%)	949.3 (+6.1%)	957.8 (+7.1%)
	var	8437.5	4250.7 (−49.6%)	4531.1 (−46.3%)	1845.3 (−78.1%)
	execTime	0.002	0.1838 (+8085%)	0.1834 (+8068%)	0.1810 (+7960%)
$\mathcal{A}_3$	avg	160.6	141.3 (−12.0%)	140.5 (−12.6%)	230.6 (+43.6%)
	var	461.4	372.3 (−19.3%)	392.5 (−14.9%)	669.2 (+45.0%)
	execTime	0.045	3.7212 (+8111%)	3.7215 (+8112%)	3.7509 (+8177%)

maximum of 1000 and the y-axis depicts the density distribution of the data plotted in R using the histogram function. Figure 4 shows that although both ForestCV algorithms can at times perform worse than RPNI, ForestMV is able to consistently outperform the RPNI. Furthermore, Fig. 4a and 4c both follow the curve for α from Fig. 1b, i.e., α is not chosen optimal. Thus, it can be assumed, that when α would be further increased, the results of the forests in Fig. 4a and 4c may be better. Table 2 analyzes the plots for Mealy machines in a similar way as Table 1 for DFAs. The observed metrics are: average (avg), variance (var) and the execution time in seconds (execTime) of each algorithm and comparing them to the RPNI. ForestMV has in Fig. 4c an average increase in correctly translated words of 43.6%, which is the highest increase of our algorithms when only considering the correct classification or translation of words. Contrarily to the results of the DFAs, the variance for the algorithms is not always lower than for RPNI. The row execTime shows, that the forest algorithms have an average increase in execution time of the factor ≈ 78.3. For both analysis, DFAs and Mealy machines, our experiments show that the execution time of ForestCV and ForestMV correlates to $\alpha \cdot m \cdot N$, where N is the execution time of the RPNI algorithm.

The experiments are executed on:

- $\mathcal{A}_1$: Intel®Core™i7-9700K CPU @ 3.60GHz processor with 32GB RAM;
- $\mathcal{A}_2$: Intel®Core™i5-7300HQ CPU @ 2.50GHz processor with 8GB RAM;
- $\mathcal{A}_3$: AMD Ryzen 7 3800X 8-Core CPU @ 3.90 GHz processor with 32GB RAM.

5 Conclusion

Classic automata learning aims at exact results which is guaranteed when a characteristic set of observations is available. However, this is often not the case, especially for practical and complex CPS. For this reason, the concepts of ForestCV and ForestMV are introduced. Both approaches show that passive learning of automata can be improved using forest structures when the characteristic set is not sampled. ForestCV does this by picking an automaton which performs best given a chosen metric. The output automaton is therefore dependent on a single automaton in the forest. By optimizing the chosen metric, ForestCV performs comparable to RPNI in our experiments, but with a lower variance. With ForestMV's approach the output is not dependent on a single Mealy machine inside the forest and hence exploits its forest structure more than ForestCV. ForestMV shows improvements for passive learning in our experiments, because it outperforms the RPNI continuous. Using ForestMV yields up to 43% more accurate results, when compared to the RPNI algorithm. This is a success in the field of passive learning, if a more accurate result is of higher significance, than the resulting longer computation time to build the forests. This might be the case for the implementation of a monitoring tool for CPS where a pre-learned model with high accuracy is required. Future directions in research for automata forests include an analysis of the influence of noise on the training set.

Acknowledgements This work is partially funded by BMBF project AGenC no. 01IS22047A.

References

1. Adriaans, P., Jacobs, C.: Using MDL for grammar induction. International Colloquium on Grammatical Inference: Algorithms and Applications (ICGI). pp. 293–306 (2006)
2. Adusumilli, S., Bhatt, D., Wang, H., Bhattacharya, P., Devabhaktuni, V.: A low-cost INS/GPS integration methodology based on random forest regression. Expert Syst. Appl. **40**(11), 4653–4659 (2013)
3. Angluin, D.: Learning regular sets from queries and counterexamples. Inf. Comput. **75**, 87–106 (1987)
4. Breiman, L.: Random forests. Mach. Learn. **45**(1) (2001)
5. Cootes, T.F., Ionita, M.C., Lindner, C., Sauer, P.: Robust and accurate shape model fitting using random forest regression voting. European Conference on Computer Vision (ECCV). pp. 278–291 (2012)
6. Gold, M.: Language identification in the limit. Inf. Control **10**(5), 447–474 (1967)
7. Groz, R., Simao, A., Petrenko, A., Oriat, C.: Inferring finite state machines without reset using state identification sequences. International Conference on Testing Software and Systems (ICTSS). pp. 161–177 (2015)
8. de la Higuera, C.: Grammatical Inference: Learning Automata and Grammars, chap. Informed learners, pp. 237–268. Cambridge University Press (2010)
9. Isberner, M., Howar, F., Steffen, B.: The open-source learnlib. International Conference on Computer Aided Verification (CAV). pp. 487–495 (2015)

10. Jourdan, G.V., Ural, H., Yenigün, H.: Reduced checking sequences using unreliable reset. Inf. Process. Lett. **115**, 532–535 (2015)
11. Maier, A.: Online passive learning of timed automata for cyber-physical production systems. International Conference on Industrial Informatics (INDIN). pp. 60–66 (2014)
12. Merten, M.: Active automata learning for real life applications. Ph.D. thesis, Technische Universität Dortmund (January 2013)
13. Mordeson, J., Malik, D.: Fuzzy automata and languages: Theory and applications. Chapman and Hall/CRC (2002)
14. Murphy, K.P., et al.: Passively learning finite automata. Tech. rep, Santa Fe Institute (1995)
15. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2022) https://www.R-project.org/
16. Schaffer, C.: Machine Learning, chap. 13, pp. 135–143. Kluwer Academic Publishers (1993)
17. Smarra, F., Jain, A., de Rubeis, T., Ambrosini, D., D'Innocenzo, A., Mangharam, R.: Data-driven model predictive control using random forests for building energy optimization and climate control. Appl. Energy **226**, 1252–1272 (2018)
18. Urbat, H., Schröder, L.: Automata learning: An algebraic approach. Symposium on Logic in Computer Science (LICS). pp. 900–914 (2020)

Domain Knowledge Injection Guidance for Predictive Maintenance

Lameya Afroze, Silke Merkelbach, Sebastian von Enzberg and Roman Dumitrescu

Abstract

With the integration of Industry 4.0 technologies, overall maintenance costs of industrial machines can be reduced by applying predictive maintenance. Unique challenges that often occur in real-time manufacturing environments require the use of domain knowledge from different experts. However, there is hardly any guidance that suggests data scientists how to inject knowledge from predictive maintenance use cases in machine learning models. This paper addresses this lack and presents a guidance for the injection of domain knowledge in machine learning models for predictive maintenance by analyzing 50 use cases from the literature. The guidance is based on the informed machine learning framework by von Rueden et al. [1]. Finally, the guidance gives a recommendation to data scientists on how domain knowledge can be injected into different phases of model development and suggests promising machine learning models for specific use cases. The guidance is applied exemplarily to two predictive maintenance use cases.

L. Afroze · S. Merkelbach (✉) · S. von Enzberg · R. Dumitrescu
Fraunhofer IEM, Paderborn, Germany
e-mail: silke.merkelbach@iem.fraunhofer.de

L. Afroze
e-mail: lameya.afroze@iem.fraunhofer.de

S. von Enzberg
e-mail: sebastian.von.enzberg@iem.fraunhofer.de

R. Dumitrescu
e-mail: roman.dumitrescu@iem.fraunhofer.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_8

Keywords

Domain knowledge • Machine learning • Predictive maintenance • Industrial data analytics • Sensor data

1 Introduction

With the rise of Industry 4.0, machine data became available and could be used for optimization in multiple use cases. One of them is predictive maintenance (PdM) which can reduce maintenance costs by up to 60% [2]. PdM is focused on condition monitoring and relies on machine learning (ML) models to decrease system downtime. However, in an industrial environment, the performance of ML models mainly depends on the quantity and quality of the available data. Raw data often lacks integrity due to different reasons, such as missing or corrupted values and noisy data. To overcome these challenges, domain knowledge is frequently integrated into ML models [3]. We will refer to this as domain knowledge injection since ML models are the basis and the knowledge enhances them.

Domain Knowledge is mostly available from domain experts in several forms. It can be injected in multiple ways based on its purpose in different phases of model development. It is often not simple to decide, which form of knowledge can be injected in which phase of the modeling process into which ML model. To find a solution for this problem, we analyzed 50 PdM use cases from the literature in which domain knowledge was injected into ML models. Von Rueden et al. [1] have proposed the Informed Machine Learning framework, that provides a taxonomy and a survey on how prior knowledge can be integrated into learning systems. Based on their framework, we developed a decision guidance for PdM use cases to assist data scientists by recommending suitable domain knowledge injection techniques for a given PdM use case and the related knowledge. We structured the use cases based on the analytics type proposed by Steenstrup et al. [4], which is seen often in the manufacturing domain. In addition, we give a recommendation for suitable ML algorithms that where applied successfully in the 50 analyzed use cases. These algorithms are ranked by the frequency they were applied. Trying the algorithms in the suggested order might lead to a good model faster.

In summary, our contributions are:

1. A variation of the framework developed by Von Rueden et al. [1] that structures available forms of domain knowledge, knowledge conversion options and injection phases especially suitable for PdM use cases.
2. A knowledge base containing 50 PdM use cases looking at the analytics type, form of domain knowledge, conversion form and possible phase for the injection into a ML model.
3. Resulting from these, a guidance with a recommendation for suitable injection phases and frequently used ML algorithms for PdM use cases.

2 Related Work

There are multiple works where domain knowledge was injected into ML models [5, 6]. Focusing on PdM applications, Serradilla et al. [2] proposed a methodology to incorporate knowledge but they focused on the process and did not address which form of knowledge should be used in which situation. Based on process discovery, Schuster et al. [7] proposed to utilize domain knowledge injection based on its provision time. They clustered the knowledge by its use in the different phases process discovery, development and post-processing. Von Rueden et al. [1] proposed a framework to incorporate prior knowledge in ML models. Since it is generic, we decided to create a variation of it especially suitable for PdM use cases according to what we learned from the literature study. Most of existing works are focused on how formalized knowledge can be injected into the models to enhance the performance of the applied ML models. According to our knowledge, so far there is no previous work suggesting which ML algorithms might be suitable for the injection of a specific form of domain knowledge in different use case scenarios. This equips the necessity of developing a guidance that can provide advice on how to inject domain knowledge into suitable ML algorithms for PdM.

3 Guidance Development

The goal of the article is to provide a guidance to data scientists to inject domain knowledge into ML models. It gives suggestions for suitable knowledge injection phases and possible ML algorithms based on a given analytics type and knowledge form. The guidance is developed with the results of a literature study in which 50 PdM use cases were analyzed. The use cases are structured according to a knowledge injection framework for PdM use cases, which is used to categorize domain knowledge, knowledge conversions and injection phases. First, the framework is introduced, followed by the literature study. The final result is the guidance in form of a table.

3.1 Knowledge Injection Framework

Based on the informed machine learning framework by Von Rueden et al. [1] and inspired by the Data mining methodology for engineering applications (CRISP-DMME) by Huber et al. [8], the knowledge injection framework depicted in Fig. 1 was developed. It is structured horizontally into three main levels which are described in the following. It is used as a frame of reference when developing a PdM use case to aid the injection process.

Form of domain knowledge related to the use case

The top row of Fig. 1 represents the first level. It comprises the domain knowledge which is directly related to the use case before any conversion or formalization. Von Rueden et al. [1] clustered the knowledge into three types: scientific knowledge, world knowledge and expert knowledge. As already suggested by them, the knowledge in these categories usually is formal, semi-formal, and informal, respectively. We chose to use the degree of formalization as categories in the framework to provide a more technical view. Besides, in manufacturing environments, domain knowledge is often available in an informal or a semi-formal form from domain experts rather than being found in a formalized form.

Informal Domain Knowledge has only indirect impact on constructing the intelligent model, since a formalization in some way is necessary. Usually, this knowledge is acquired by experts through their working life such as simple heuristics or intuitions. Data scientists get benefits from this knowledge to make decisions by following different standard techniques. Examples from PdM use cases include different types of sensors that are suitable to collect a specific type of data, their location or the total number of sensors. Informal domain knowledge is also helpful to apply various pre-processing techniques to the data, such as normalizing/filtering data, or applying different physics-based pre-processing techniques like Fast Fourier Transform, Short-time Fourier Transform, or Wavelet Transform to time series data, which have proven to be useful in similar use cases. This special form of informal domain knowledge can further be defined as domain-specific data science knowledge.

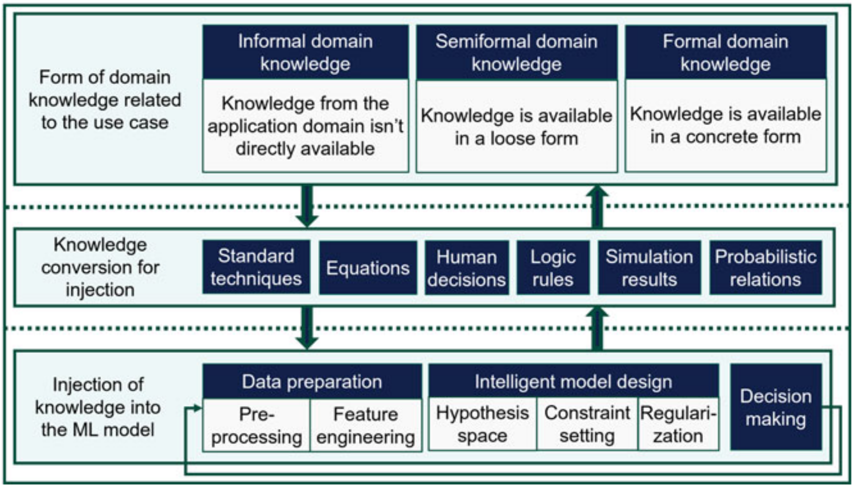

Form of domain knowledge related to the use case
Informal domain knowledge
Knowledge from the application domain isn't directly available
Semiformal domain knowledge
Knowledge is available in a loose form
Formal domain knowledge
Knowledge is available in a concrete form
Knowledge conversion for injection
Standard techniques
Equations
Human decisions
Logic rules
Simulation results
Probabilistic relations
Injection of knowledge into the ML model
Data preparation
Pre-processing
Feature engineering
Intelligent model design
Hypothesis space
Constraint setting
Regulari-zation
Decision making

Fig. 1 Knowledge injection framework for PdM use cases

Semi-formal Domain Knowledge is more structured and explicitly available. But it is not entirely structured and needs to be formalized before injection. In the analyzed use cases, it was mainly injected by using logic rules to process data based on an expert opinion or expert-defined thresholds to filter/label input data.

Formal Domain Knowledge refers to expert knowledge that can be injected in a standardized form. Examples include simulation-generated machine data which can be used for model training or equations defined by the domain expert to design the model architecture.

Knowledge conversion for injection

The knowledge conversion is carried out at the second level depicted in the middle row in Fig. 1. It acts as a bridge between domain experts and data scientists. In the literature, the knowledge was converged into a variety of formats, which include the following:

- Standard techniques which proved to have a good performance in similar use cases in the past, such as specific pre-processing techniques.
- Different forms of equations, such as mass balances.
- Human decisions, such as a rating if the machine sounds normal.
- Logic rules, such as the ones contained in a programmable logic controller.
- Simulation results, for example from a digital twin of a machine.
- Probabilistic relations, such as the Weibull distribution [9] to fit the training model.

Injection of Knowledge into the ML Model

The bottom row of Fig. 1 represents the third level. It shows possible phases for the injection of domain knowledge during model development. This level is structured into three groups: data preparation, intelligent model design, and decision making. In data preparation, converged knowledge is used to pre-process the data and to generate relevant features through feature engineering. The next group aims to enhance the performance of an ML model by injecting knowledge into three different tasks of the intelligent model design phase, which are: hypothesis space definition, constraint setting, and loss function regularization. The last group is decision making in which a component to make automated decisions based on the prediction of the ML model is developed. It is mainly used to perform prescriptive tasks such as making appropriate maintenance decisions to mitigate the impact of unpleasant incidents.

Due to the variety of knowledge, it is not possible to give standard paths through the framework. Also, not all of the levels apply in all use cases. Especially for formal knowledge, the conversion might not be needed since the knowledge often can be used directly. Some of the phases in the third level can be used in combination with all conversions, others were found to be applied only in specific phases. All three levels are linked, providing the possibility to switch the levels when further insights into the use case are gained. For example, a standard technique can be used to create results on which an expert decision can be made to apply further conversion techniques.

3.2 Literature Study and Construction of the Knowledge Base

We conducted a literature study, resulting in the analysis of 50 PdM use cases. Therefore, we searched for PdM use cases with the terms ‘condition monitoring’, ‘fault detection’, ‘fault diagnosis’, ‘predictive maintenance’, and ‘prescriptive maintenance’ in combination with the terms ‘artificial intelligence’, ‘machine learning’, ‘domain knowledge’, and ‘domain knowledge integration’. We searched in Google Scholar, ScienceDirect, Springer, IEEE Explore, ACM Digital Library, and Elsevier for works published mainly between 2017 and 2022.

We clustered the use cases according to their objective. Therefore, we used the following four analytics types [4]:

- Descriptive analytics provides information on what has happened
- Diagnosis explains why something has happened and the reasons behind it
- Predictive analytics gives an estimate of what will happen in the future
- Prescriptive analytics offers recommendations on how to influence the future

Each of these four clusters was further divided by the form of domain knowledge according to the framework shown in Fig. 1. Table 1 depicts the overall knowledge base, showing which conversion of knowledge was injected in which phase for each of the 50 use cases. The table shows, that the main conversions for knowledge injection in the analyzed use cases were the application of standard techniques and human decisions, followed by equations. Logic rules, simulation results and probabilistic relations were applied in only a few use cases. Looking at the third level of the framework, the most relevant phase for injection is pre-processing, followed by feature engineering and hypothesis space. Constraint setting, regularization and decision making were not applied in many of the analyzed use cases. The analytics type prescriptive has far less use cases than the rest. Also, not all forms of knowledge are covered for all use cases.

3.3 Guidance Creation

The guidance is shown in Table 2. It takes the analytics type and knowledge form as input and gives suggestions for a suitable injection phase and suitable ML algorithms. The injection phases and ML algorithms are ranked based on their number of occurrence in the analyzed use cases. We recommend to try the suggestions of the guidance based on the ranking of the entries. For example, if data scientists are searching for a suggestion for a diagnosis use case with informal domain knowledge, trying feature engineering in combination with support vector machine, is the most promising way. In case this does not provide sufficient results, the other ML algorithms could be tried in the given order. Afterwards, the injection technique might be changed to pre-processing. It is also possible to apply different forms of

Table 1 Knowledge base containing 50 PdM use cases structured by analytics type and according to the knowledge injection framework. (ST= standard techniques, EQ= equations, LR= logic rules, SR= simulation results, PR=probabilistic relations, HD= human decisions; PP= pre-processing, FE= feature engineering, HS= hypothesis space, CS= constraint setting, RZ= loss function regularization, DM= decision making)

Analytics type	Form of domain	Knowledge conversion for injection						Injection of knowledge into the ML model					
	knowledge	ST	EQ	LR	SR	PR	HD	PP	FE	HS	CS	RZ	DM
Descriptive	Informal	[10–13]	[12, 14]				[10, 12–14]	[11–13]	[10, 13, 14]	[13]	[13]		
	Semi-formal	[15, 16]	[15]				[15]	[15, 16]				[15]	
Diagnosis	Informal	[14], [17–24]	[17, 18, 21, 25, 26]	[20]			[14, 18, 20, 22–28]	[14, 17, 18, 20–24]	[14, 18–21, 25–27]	[17, 18, 23, 26–28]	[28]		[14]
	Semi-formal	[29–32]	[31–35]	[36]			[29, 32–36]	[29–32, 35, 36]	[30, 31, 33, 36]	[31, 34]	[34]	[30]	
Predictive	Informal	[2, 37–47]	[41, 47]				[2, 37–39, 41, 42, 45–48]	[2, 37–47]	[2, 37, 38, 42, 44]	[2, 37], [39, 41, 45, 47, 48],	[2]		
	Semi-formal	[49]	[9, 49]			[9]	[9]	[9, 49]	[9, 49]				
	Formal	[45, 46, 50]	[35]		[45, 46, 50, 51]		[35, 45, 46, 50]	[35, 45, 46, 50, 51]		[45, 50]			
Prescriptive	Semi-formal	[52]		[52]				[52]					[52]
	Formal			[53]	[53]	[53]		[53]					[53]

Table 2 Decision guidance suggesting suitable knowledge injection phase and ML algorithms based on a given analytics type and knowledge form. For each ML algorithm, one example use case is provided where the model was applied as an orientation. The number of occurrence of the ML algorithms in the analyzed use cases is given in parenthesis if it is greater than one. (PP= pre-processing, FE= feature engineering, HS= hypothesis space, CS= constraint setting, RZ= loss function regularization, DM= decision making)

Input from the user		Recommendations from the guidance	
Analytics type	Knowledge form	Suitable knowledge injection phase	Top 5 ML algorithms according to the number of occurrence
Descriptive	Informal	FE, PP, CS, HS	TSAK-SVM [10], SAE [15], MLP [41], Kmeans [14], SVM [26]
	Semiformal	PP, RZ	SAE, PCA [42], SVM, GA [13]
Diagnosis	Informal	FE, PP, HS, DM, CS	SVM(12), ANN[25](4), RF[2](3), KNN[41](3), BPNN[48](3)
	Semiformal	PP, FE, HS, CS, RZ	SVM, ANN, RF, KNN, BPNN
Predictive	Informal	PP, HS, FE, CS	SVR [44](5), SVM(5), LSTM [46](5), MLP(4), RF(4)
	Semiformal	PP, FE	SVR, MLP, LSTM, SVM, RF
	Formal	PP, HS	SVR, SVM, MLP, LSTM, RF
Prescriptive	Semiformal	PP, DM	DPCA, Kmeans
	Formal	PP, DM	HMM

knowledge in multiple modeling phases. Therefore, the guidance should be followed once for each type of available knowledge.

We have used the guidance as a knowledge base for a prototype of a recommender system. The prototype is available at GitHub[1].

[1] https://github.com/lameya116/Domain_knowledge_injection_guidance.git

4 Examples for the Application of the Guidance

In the following we show how the guidance may be used when applied to two real-world use-cases. The first one is a diagnostic use case that monitors the condition inside a disk stack separator [54]. Domain knowledge was available in a semi-formal form which was converted into standard techniques and logic rules at the knowledge conversion level. From the guidance suggestions, pre-processing, feature engineering, and hypothesis space definition were applied to develop the ML model. Standard techniques and logic rules were applied to pre-process the data, and then a human decision was taken to detect suitable features and set a hypothesis space. Among the models suggested by the guidance, the highest accuracy was achieved by random forest (RF), which is 91.27%.

The second use case is a predictive use case where we used the NASA IMS bearing dataset [55] to predict outer race defects of bearing 1 to estimate its remaining useful life. Here, the domain knowledge was available in an informal form. Pre-processing, feature engineering, and hypothesis space definition were used as knowledge injection phases as suggested by the guidance. In the knowledge conversion level, FFT was used as a standard technique to transform informal knowledge and perform pre-processing. Besides, human decisions were made to conduct feature engineering and define the hypothesis space. Among the suggested models, SVR performed best with a root mean square error of 0.12.

5 Discussion

The evaluation has shown that, by using the guidance, it is possible to quickly select suitable knowledge injection phases and ML algorithms for PdM use cases. It does not guarantee that the suggestions will be the best as the performance of any ML model also depends on the underlying systems and data, including their quality, quantity and type. Choosing one over another mainly depends on the users with regard to the selected use case. However, the users can get ideas and/or a starting point on how to create models for PdM use cases with the injection of domain knowledge. The guidance assists its users both in a direct and indirect way. Directly, the users get suggestions of suitable phases and probably suitable models. Indirectly, the users can get additional ideas from the knowledge injection framework on different further possibilities to inject domain knowledge. At this moment, the knowledge base lacks sufficient use case examples for some of the criteria. For instance, prescriptive analytics contains fewer use cases as the current literature contains fewer examples in this area. This limits the guidance's ability to give reliable suggestions for this type of use case. Also, the knowledge base has not covered examples for each possible form of domain knowledge and has not the same number of use cases for the conversions. To overcome these problems and to improve the guidance suggestions, it is required to analyze more use cases.

6 Conclusion

This paper presents a guidance which assists its users to inject domain knowledge in ML models for PdM use cases. It suggests possible injection phases and suitable ML algorithms with the analytics type and the form of domain knowledge as input. It is based on a literature study, in which 50 PdM use cases were analyzed. The recommendation of the guidance has been applied to two PdM use cases and delivered a good result in both cases. In its current form, the underlying knowledge base does not contain a balanced amount of use cases in every category. For more reliable suggestions, more use cases should be investigated in the future which are related to the lesser represented categories.

Acknowledgements Supported by the German Federal Ministry for Economic Affairs and Climate Action under grant 03EN4004B.

References

1. Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Walczak, M., Pfrommer, J., Pick, A. et al.: Informed machine learning-a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Transactions on Knowledge and Data Engineering. (2021)
2. Serradilla, O., Zugasti, E., Ramirez de Okariz, J., Rodriguez, J., Zurutuza, U.: Methodology for data-driven predictive maintenance models design, development and implementation on manufacturing guided by domain knowledge. Int. J. Comput. Integr. Manuf. 1–25. (2022)
3. Kong, W., Qiao, F., Wu, Q.: Real-manufacturing-oriented big data analysis and data value evaluation with domain knowledge. Computat. Stat. **35**(2), 515–538 (2020)
4. Steenstrup, K., Sallam, R., Eriksen, L., Jacobson, S.: Industrial analytics revolutionizes big data in the digital business. Gartner Res. (2014)
5. Olmos-Sánchez, K., Rodas-Osollo, J.: Knowledge management for informally structured domains: Challenges and proposals. Knowledge Management Strategies and Applications, pp. 85–102. IntechOpen, London (2017)
6. Dash, Tirtharaj: Chitlangia, Sharad, Ahuja, Aditya, Srinivasan, Ashwin: A review of some techniques for inclusion of domain-knowledge into deep neural networks. Sci. Rep. **12**(1), 1–15 (2022)
7. Schuster, D., van Zelst, S.J., van der Aalst, W.MP.: Utilizing domain knowledge in data-driven process discovery: A literature review. Comput. Ind. **137**, 103612 (2022)
8. Huber, S., Wiemer, H., Schneider, D., Ihlenfeldt, S.: Dmme: Data mining methodology for engineering applications-a holistic extension to the crisp-dm model. Procedia Cirp **79**, 403–408 (2019)
9. Ben Ali, J., Chebel-Morello, B., Saidi, L., Malinowski, S., Fnaiech, F.: Accurate bearing remaining useful life prediction based on weibull distribution and artificial neural network. Mech. Syst. Sign. Process. **56**, 150–172 (2015)
10. Zhu, F., Jia, X., Miller, M., Li, X., Li, F., Wang, Y., Lee, J.: Methodology for important sensor screening for fault detection and classification in semiconductor manufacturing. IEEE Trans. Semicond. Manuf. **34**(1), 65–73 (2020)

11. Gao, Q., Tang, H.-S., Xiang, J.-W., Zhong, Y.: A multi-sensor fault detection strategy for axial piston pump using the walsh transform method. Int J Distrib. Sens. Netw. **14**(4), 1550147718772531 (2018)
12. Wen, X., You, L.: A novel rolling bearing fault detection method based on wavelet transform and empirical mode decomposition. In 2019 Chinese Control Conference (CCC), pp. 5024–5027. IEEE (2019)
13. Jalali, S.K., Ghandi, H., Motamedi, M.: Intelligent condition monitoring of ball bearings faults by combination of genetic algorithm and support vector machines. J. Nondestr. Eval. **39**(1), 1–12 (2020)
14. Uhlmann, E., Pastl Pontes, R., Geisert, C., Hohwieler, E.: Cluster identification of sensor data for predictive maintenance in a selective laser melting machine tool. Procedia Manuf. **24**, 60–65, (2018)
15. Liang, X., Duan, F., Bennett, I., Mba, D.: A sparse autoencoder-based unsupervised scheme for pump fault detection and isolation. Appl. Sci. **10**(19), 6789 (2020)
16. Germán-Salló, Z., Strnad, G.: Signal processing methods in fault detection in manufacturing systems. Procedia Manuf. **22**, 613–620 (2018)
17. Liu, H., Li, L., Ma, J.: Rolling bearing fault diagnosis based on stft-deep learning and sound signals. Shock Vib. **2016** (2016)
18. Serradilla, O., Zugasti, E., de Okariz, J.R., Rodriguez, J., Zurutuza, U.: Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data. Appl. Sci. **11**(16), 7376 (2021)
19. Ejike Akpudo, U., Hur, J.-W.: Intelligent solenoid pump fault detection based on mfcc features, lle and svm. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 404–408. IEEE (2020)
20. Raghu Vamsi, P., Chahuan, A.: Machine learning based hybrid model for fault detection in wireless sensors data. EAI Endorsed Trans. Scalable Inf. Syst. **7**(24) (2020)
21. Indriawati, K., Widjiantoro, B.L. et al.: Discriminant feature extraction of motor current signal analysis and vibration for centrifugal pump fault detection. In 2021 International Conference on Instrumentation, Control, and Automation (ICA), pp. 207–212. IEEE (2021)
22. Zawad Ali, M., Nasmus Sakib Khan Shabbir, Md., Liang, X., Zhang, Y., Hu, T.: Machine learning-based fault diagnosis for single-and multi-faults in induction motors using measured stator currents and vibration signals. IEEE Trans. Ind. Appl. **55**(3), 2378–2391 (2019)
23. Yu, M., Xiao, C., Liu, M.J., Chen, J., Yuan, R.: Motor fault diagnosis based on teaching and learning gray wolf algorithm optimized support vector machine. J. PhyS. Conf. Ser. **2232**, 012017. (IOP Publishing) (2022)
24. Hosseinpoor, Z., Mehdi Arefi, M., Razavi-Far, R., Mozafari, N., Hazbavi, S.: Virtual sensors for fault diagnosis: a case of induction motor broken rotor bar. IEEE Sens. J. **21**(4), 5044–5051 (2020)
25. Singh Ranawat, N., Kumar Kankar, P., Miglani A.: Fault diagnosis in centrifugal pump using support vector machine and artificial neural network. J. Engg. Res. EMSME Special Issue pp. **99**, 111 (2021)
26. Zhu, X., Xiong, J., Liang, Q.: Fault diagnosis of rotation machinery based on support vector machine optimized by quantum genetic algorithm. IEEE Access **6**, 33583–33588 (2018)
27. Li, Z., Wang, Y., Wang, K.-S.: Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. Adv. Manuf. **5**(4):377–387 (2017)
28. Shao, H., Jiang, H., Lin, Y., Li, X.: A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. Mech. Syst. Signal Process. **102**, 278–297 (2018)
29. Buabeng, A., Simons, A., Kena Frempong, N., Yevenyo Ziggah, Y.: A novel hybrid predictive maintenance model based on clustering, smote and multi-layer perceptron neural network optimised with grey wolf algorithm. SN Appl. Sci. **3**(5):1–24 (2021)

30. Turanoglu Bekar, E., Nyqvist, P., Skoogh, A.: An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. Adv. Mech. Eng. **12**(5):1687814020919207 (2020)
31. Saud ALTobi, M.A., Bevan, G., Wallace, P., Harrison, D., Ramachandran, K.P.: Fault diagnosis of a centrifugal pump using mlp-gabp and svm with cwt. Eng. Sci. Techn. Int. J. **22**(3), 854–861 (2019)
32. Siano, D., Panza, M.A.: Diagnostic method by using vibration analysis for pump fault detection. Energy Procedia **148**, 10–17 (2018)
33. Irfan, M., Saeed Alwadie, A., Glowacz, A., Awais, M., Rahman, S., Asif Khan, M.K., Jalalah, M., Alshorman, O., Caesarendra, W.: A novel feature extraction and fault detection technique for the intelligent fault identification of water pump bearings. Sensors **21**(12), 4225 (2021)
34. Shao, H., Jiang, H., Zhao, H., Wang, F.: A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. Mech. Syst. Signal Process. **95**, 187–204 (2017)
35. Rengasamy, D., Jafari, M., Rothwell, B., Chen, X., Figueredo, G.P.: Deep learning with dynamically weighted loss function for sensor-based prognostics and health management. Sensors **20**(3), 723 (2020)
36. Jiang, W., Chen, Z., Zhang, N., Xu, Y.: A hybrid fault diagnosis method for rolling bearing with improved multiscale dispersion entropy-based state pre-detection and feature fusion. In 2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing), pp. 1–4. IEEE (2021)
37. Soualhi, A., Medjaher, K., Zerhouni, N.: Bearing health monitoring based on hilbert-huang transform, support vector machine, and regression. IEEE Transactions on instrumentation and measurement **64**(1), 52–62 (2014)
38. Sui, W., Zhang, D., Qiu, X., Zhang, W., Yuan, L.:. Prediction of bearing remaining useful life based on mutual information and support vector regression model. In IOP Conference Series: Materials Science and Engineering, **533**, 012032. IOP Publishing (2019)
39. Nasser, A. Al-Khazraji, H. A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance. Int. J. Electr. Comput. Eng. **12**(1), 2088–8708 (2022)
40. Kanawaday, A., Sane, A.: Machine learning for predictive maintenance of industrial machines using iot sensor data. In 2017 8th IEEE international conference on software engineering and service science (ICSESS), pp. 87–90. IEEE (2017)
41. Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., Wang, J.: Machine health monitoring using local feature-based gated recurrent unit networks. IEEE Trans. Ind. Electr. **65**(2), 1539–1548 (2017)
42. Chazhoor, A., Mounika, Y., Vergin Raja Sarobin, M., Sanjana, M.V., Yasashvini, R.: Predictive maintenance using machine learning based classification models. IOP Conference Series: Mater. Sci. Eng. **954**, 012001. IOP Publishing (2020)
43. Binding, A., Dykeman, N., Pang, S.: Machine learning predictive maintenance on data in the wild. In 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), pp. 507–512. IEEE (2019)
44. Akuruyejo, M., Kowontan, S., Ben Ali, J.: A data-driven approach based health indicator for remaining useful life estimation of bearings. In 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), pp. 284–289. IEEE (2017)
45. Malhotra, P., Tv, V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder. arXiv preprint arXiv:1608.06154 (2016)
46. Zheng, S., Ristovski, K., Farahat, A., Gupta, C.: Long short-term memory network for remaining useful life estimation. In 2017 IEEE international conference on prognostics and health management (ICPHM), pp. 88–95. IEEE. (2017)

47. Kabir, F., Foggo, B., Yu, N.: Data driven predictive maintenance of distribution transformers. In 2018 China International Conference on Electricity Distribution (CICED), PP. 312–316. IEEE. (2018)
48. Nikfar, M., Bitencourt, J., Mykoniatis, K.: A two-phase machine learning approach for predictive maintenance of low voltage industrial motors. Procedia Comput. Sci. **200**, 111–120 (2022)
49. Sampaio, G.S., de Aguiar Vallim Filho, A.R., da Silva, L.S., da Silva, L.A.: Prediction of motor failure time using an artificial neural network. Sensors **19**(19):4342 (2019)
50. Shcherbakov, M., Sai, C.: A hybrid deep learning framework for intelligent predictive maintenance of cyber-physical systems. ACM Transactions on Cyber-Physical Systems (TCPS) **6**(2), 1–22 (2022)
51. Yoon, A.S., Lee, T., Lim, Y., Jung, D., Kang, P., Kim, D., Park, K., Choi, Y.: Semi-supervised learning with deep generative models for asset failure prediction. arXiv:1709.00845
52. Wenjin, Y., Dillon, T., Mostafa, F., Rahayu, W., Liu, Y.: A global manufacturing big data ecosystem for fault detection in predictive maintenance. IEEE Trans. Ind. Inf. **16**(1), 183–192 (2019)
53. Zhao, X., Kang, Y., Yan, H., Ju, F.: Semi-supervised constrained hidden markov model using multiple sensors for remaining useful life prediction and optimal predictive maintenance. Annual Conference of the PHM Society, **11** (2019)
54. Merkelbach, S., Afroze, L., Janssen, N., von Enzberg, S., Kühn, A., Dumitrescu, R.: Using vibration data to classify conditions in disk stack separators. Vibroengineering PROCEDIA **46**, 21–26 (2022)
55. Lee, J., Qiu, H., Yu, G., Lin, J.: Ims bearing run-to-failure dataset. (2007)

Towards a Systematic Approach for Prescriptive Analytics Use Cases in Smart Factories

Julian Weller, Nico Migenda, Rui Liu, Arthur Wegel,
Sebastian von Enzberg, Martin Kohlhase, Wolfram Schenck,
and Roman Dumitrescu

The original version of the chapter has been revised. A correction to this chapter can be found at https://doi.org/10.1007/978-3-031-47062-2_13

J. Weller (✉) · A. Wegel · S. von Enzberg
Fraunhofer Institute for Mechatronic Systems Design, Paderborn, Germany
e-mail: julian.weller@iem.fraunhofer.de

A. Wegel
e-mail: arthur.wegel@iem.fraunhofer.de

S. von Enzberg
e-mail: sebastian.von.enzberg@iem.fraunhofer.de

N. Migenda · R. Liu · M. Kohlhase
Faculty of Engineering and Mathematics, Bielefeld University of Applied Sciences, Gütersloh, Germany
e-mail: nico.migenda@fh-bielefeld.de

R. Liu
e-mail: rui.liu@fh-bielefeld.de

M. Kohlhase
e-mail: martin.kohlhase@fh-bielefeld.de

W. Schenck
CfADS, Faculty of Engineering and Mathematics, Bielefeld University of Applied Sciences, Gütersloh, Germany
e-mail: wolfram.schenck@fh-bielefeld.de

R. Dumitrescu
Fraunhofer IEM and Heinz Nixdorf Institute, Paderborn, Germany
e-mail: roman.dumitrescu@iem.fraunhofer.de

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien
für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_9

Abstract

Manufacturing systems are dynamic and exhibit increasing complexity and uncertainty. Smart manufacturing uses Data Analytics methods to optimize manufacturing processes, systems and products. One approach to structure use cases in production management in smart manufacturing is the Product-Process-Resource (PPR) model, where the resource executes a process on a given product. The PPR model needs to be extended for smart manufacturing, to meet the requirements of prescriptive analytics (but not exclusively). Our contributions are an extended PPR model for prescriptive analytics (P2PR) that involves environmental effects, expert knowledge and adds a process sub-model distinguishing between manufacturing and supervisory processes. We develop prescriptive analytics decision-making categories based on the area of validity and the degree of interconnectivity. The combination results in a systematization scheme for prescriptive analytics use cases in a smart factory environment. It assists entities to find shared characteristics in different prescriptive smart factory use cases within one production ecosystem. A mapping of prescriptive algorithms (as part of a use case) to a category and domain is enabled for future case studies.

Keywords

Prescriptive Analytics • Use Case Systematization • Smart Factory • PPR Model • P2PR Model

1 Introduction

Today's customer demands, e.g. high product quality or sustainable production at low cost, impose high requirements on manufacturing companies. Digitalization of processes enables industries to increase process stability, product quality, reduce downtime, and increasing throughput. This leads to highly complex production systems with interconnection of hard- and software. Manufacturing companies must balance supervision and orchestration of resources, executing processes for high quality products.

Big Data is leveraged by Data Analytics, allowing for real-time insight, process optimization, and the prediction of failures before they impact corresponding operations [1]. Data Analytics can be divided into descriptive, diagnostic, predictive and prescriptive approaches. The most advanced form of Data Analytics is prescriptive analytics, which offers the highest degree of decision automation by providing the best possible reaction strategies. Prescriptive Analytics can develop its full potential when applied to the entire production management system [1]. While the combination of production system with data-driven analytics towards smart manufacturing seems very promising, propagation is very low. We do spectate that Data Analytics and AI is almost not used in small and medium-sized enterprises (SME) across the EU [2]. The successful Data Analytics application to solve manufacturing challenges usually gets as far as the predictive level; prescriptive approaches still face manifold challenges.

Companies usually lack a mature data infrastructure to apply AI in production, even though this is an essential part of deployment [3]. Use cases are implemented as lighthouse projects, but a lack of methodology and widespread use is still prevalent [4]. This is due to a lack of transparency on return on investment when implementing such complex solutions [2]. Manufacturing companies unique value proposition revolves around their domain knowledge. Extensive knowledge and capacity for the implementation of complex AI models is often lacking [5].

Prescriptive Analytics needs a full understanding of all elements in the feedback loop of the System of Interest (SoI) [6]. Possible SoIs are resources, processes and products with respective use cases like Prescriptive Maintenance, Autonomous Processes, Prescriptive Quality. Influences on the SoI are derived from inputs and environmental influences, with environmental production resources and processes leading to a complex System of Systems. This creates a highly complex solution space with a lot of factors of uncertainty [6]. Thus, we identify a need to structure use cases to improve a further understanding of the process of developing successful Prescriptive Analytics solutions (in all technical readiness levels). A structuring approach enables data scientist to further standardize and reuse developed artifacts, thus decreasing development time of analytics and increasing acceptance and propagation of use cases.

While approaches like reinforcement learning allow completely data-driven decision making and prescription, it involves knowledge about a SoI in context of its environment and a system-of-systems-understanding. However, current development of Data Analytics solutions is mainly focused on a singular system of interest. A simplified approach is the formalizing of expert knowledge into standard reaction strategies.

Based on the state of the art (Chap. 2), a research gap regarding the characterization of prescriptive analytics use cases in smart manufacturing is identified through literature research. Chapter 3 addresses these issues by presenting an approach to combine both analytics and smart manufacturing driven perspectives on the use case domain. The approach was derived through the extension and refitting of existing approaches. The proposed extended P2PR-model for prescriptive analytics in combination with the provided performance levels enable a new standard in smart factories for reaction strategies and decision making. The summary, future work and further research gaps based on the results are provided in Chap. 4 (conclusion and outlook).

The research goal of this paper is to present approaches to enable SMEs to structure prescriptive analytics use cases in smart factory environments. Our contributions compared with existing work are:

- We extend the existing PPR-models towards prescriptive analytics and provide a systematic approach to classify use cases and analytics levels.
- We interlink the degree of decision support and expert knowledge to create performance levels for prescriptive analytics in smart manufacturing systems.
- We propose a 3-dimensional approach to structure future use cases and their algorithms to find similarities and extend the reusability and applicability of future prescriptive analytics case studies.

2 State of the Art

Manufacturing analytics describes the use of data-based algorithms to ensure product quality, reduce maintenance costs, and optimize production processes. Manufacturing analytics methods are divided into four levels, with each level further reducing the need for human interaction [7]. The levels descriptive, diagnostic, predictive, and prescriptive Analytics will be explained in the following:

The most basic form is descriptive analytics, which provides basic insight and visualizations of what has happened in previous production cycles [8, 9]. Diagnostic analytics methods close this gap by mapping data with possible causes to identify correlations between variables, sources of trends, and errors. When it is known what has happened in the past and why it happened, it is possible to predict what is likely to occur in the future. Predictive analytics refers to the use of statistical modeling, data mining techniques, and machine learning to make predictions about future outcomes based on historical and current data, such as predicting the next maintenance date [10, 11]. Prescriptive analytics builds upon the three other types of Data Analytics which describe the present and make predictions about the future [4, 12]. It processes and evaluates predictions as well as detected errors and anomalies and derives reaction strategies or recommendations for actions. In addition, expert knowledge can be introduced to further enhance this process.

Prescriptive Analytics is usually structured into a fully automated support and a human in the loop approach according to Gartner [7]. Vater et al. propose to structure existing approaches by the level of human interaction and the chosen IT infrastructure [4].

2.1 Formalization of Data Analytics Use Cases in Smart Factories

In the domain of Data Analytics, a smart factory specific systematization of use cases for prescriptive analytics has not been discovered via systematic literature research. Authors mainly focus on solving defined problems with narrow boundaries [13–17]. Use cases are either described in a process-related, result-based or artifact-based way. Workflows to recreate analytics solutions (crisp-dm) are process-related examples. Result-based examples are typical of experiment based and centered machine learning papers. Artifact-based papers develop a use case specific solution that evolved around a reusable part like a UI or algorithm. Kühn et al. propose to structure use cases regarding their interaction with algorithms, it-infrastructure, and data sources [18]. This agnostic approach does not consider which the solution has. A canvas to identify analytics use cases regarding smart services was developed by Panzner et al. [19]. The approach focuses on generating use cases.

2.2 Product, Process and Resource in Smart Factories

A "product" is defined as the final or intermediate result of production processes. The flow descriptions necessary for the creation of a product or its individual parts and assemblies are described by the process. The process describes all interacting operations that are necessary for the completion of a product. Resources are the objects necessary for the execution of a process or generally to fulfill a task (e.g., infrastructure and employees), but not the work object or the product to be produced itself. For a more detailed description of PPR, we refer to [20, 21]. The PPR model defines the links between the three entities and is deeply rooted in quality management. In [22], a skill definition of PPR is proposed. A skill is an abstract form of a process that defines the ability of a resource to perform a process. A task is the application of a skill to a defined product with a desired outcome. A similar approach for Machines as a Service is found in [23] where the product and resource are decoupled from each other. A resource has skills that must match the requirements of the product. Only on a match, a process task is executed.

3 Structuring Prescriptive Analytics in a Smart Factory Environment

All analyzed approaches regarding smart factory use cases deal with specific problem-solution-combinations and do not generalize their approach sufficiently. Solutions regarding data governance and integration are missing. Hence, to our knowledge, no method with a holistic approach towards further specifying and categorizing use cases in a smart factory environment could be identified. The general approach for dealing with uncertainty in use case specifics was identified as a research gap.

3.1 Data Analytics View on Use Cases

One possible way to structure analytics approaches is given by the analytics levels according to Gartner (see Chap. 2). The one-dimensional and algorithmic view does not deal well with the complexity and uncertainty of possible use cases in manufacturing. Furthermore, the different approaches and levels of possible prescriptive approaches are not well represented regarding different kinds of prescriptive analytics. Dimensions like decision impact, area of validity and the interconnectivity of different decision-making systems are not represented. We propose a further differentiation between the different possible kinds of prescriptive analytics.

This is motivated from a technical point of view. Autonomy levels and other Industry 4.0 concepts work in parallel to the proposed approach. The different levels are visualized in Fig. 1. They are needed to make prescriptive analytics more applicable regarding their

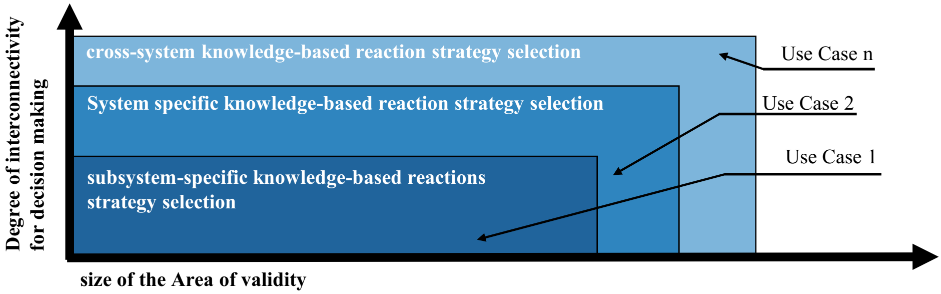

Degree of interconnectivity for decision making
cross-system knowledge-based reaction strategy selection
System specific knowledge-based reaction strategy selection
subsystem-specific knowledge-based reactions strategy selection
Use Case n
Use Case 2
Use Case 1
size of the Area of validity

Fig. 1 Vision of smart factory use cases in prescriptive analytics (based on [24])

impact on smart factory production systems. A structuring approach assists by further specifying the solution space for prescriptive analytics and the validity of the decision under observance. The degree of decision support and the degree of integrated expert knowledge in the system create the X and Y axis. The subsystem-specific strategy selection is already complex, but the first level of prescriptive analytics. Based on that, system specific reaction strategy selection exists. The highest form of prescriptive analytics makes decisions that are taken with regard to other systems' decisions as well (outside the SoI). The model enables a more differentiated discussion about different solutions for prescriptive analytics use cases. The goal is to later enable a mapping between the technical solution (algorithm), the prescriptive analytics level (Fig. 1) and use case specifics (e.g., the PPR model). It mainly focuses on the value proposition of the analytics level "prescriptive". It is contributing towards the technical feasibility of the automation of actionable decisions, by making them compatible with other concepts such as autonomy, reconfiguration, online learning (update mechanisms), self-healing mechanisms, human in the loop, and other approaches.

3.2 Smart Manufacturing View on Use Cases

A second dimension is needed to connect smart factory and Data Analytics views on use cases. The second dimension can be built based on the existing PPR model (product, process and resource) as presented in Chap. 2. The model was initially developed to structure workflows in car manufacturing companies. The existing model does not focus well on the differentiation between different kinds of processes. Manufacturing processes are processes like stamping or welding. Other processes like production planning, scheduling, logistical orchestration or the allocation of resources in production have completely different characteristics. Organizational processes and manufacturing processes share the same name but demand different analytical tools and algorithms to handle them (compare [12]). The goal is to ensure a maximum difference between different classes for analytics use cases (and used algorithms). Thus, we propose to differentiate the "process" into a supervisory and a manufacturing related one. This way, business and organizational processes and their use cases are strictly divided and differentiated from production processes like welding and stamping. Based on Stanev [21] and Haasis et al. [20], we propose the following definitions for product, process, and resource: The definition is given to further standardize the usage of all terms and lays a foundation for the extended PPR (P2PR) model:

- Process (supervisory): The process is no physical entity. It maps purpose (task) and realizer (resource) to create products. Thus, it defines the order in which resources are invested to create a product and which state of the product is regarded as a finished product. The realizer and the purpose do not have to be tied directly to a product. A Data Analytics process use case optimizes the orchestration and behavior behind the allocation of resources which are embedded in the process.
- Process (Manufacturing): Manufacturing Processes describe operations like welding, stamping or glueing. A data analytics manufacturing process use case optimizes their parameters, regulation or reconfiguration.
- Resource: Resources represent the entities, that are used to generate products with defined characteristics. Resources are orchestrated by a process and always exist in a logical relation to other resources and steps in the production value chain. allocated to a task by a process. A Data Analytics resource use case deals with maintenance, health status and behavior of the resource regarding its ability to do its intended job (ensure ability to produce under given KPIs).
- Product: Products in a smart factory environment are physical entities with characteristics (planed and realized). They are instantiated because a process demands it, and a resource fulfills the demand. A Data Analytics product use case analyzes and optimizes the quality, integrity and characteristics of a product instance or class.

Additionally, an environment can be added to integrate non-manufacturing, but use case related objects into the model. E.g., the delivery of parts from a partner can take longer, thus having an influence on the existing product, process, and resource. The new proposed model is summarized in Fig. 2. The surrounding text boxes are there to further enhance the model understanding and represent examples. Different authors [22, 23] have altered the PPR model as well but have not focused on its application in the smart factory domain (see Chap. 2). The definition is meant as a reference for the design and assignment of future use cases into a category. The model needs to be expanded in the future for non-smart factory (but enterprise) related advanced analytics use cases. The model explicitly does not include the manufacturing environment (e.g. order and purchase processes).

Our solution to combine the Data analytics domain and the smart factory environment is to merge both structuring views to generate a general use case structuring approach for smart manufacturing Data analytics related use cases. The goal is to group use cases

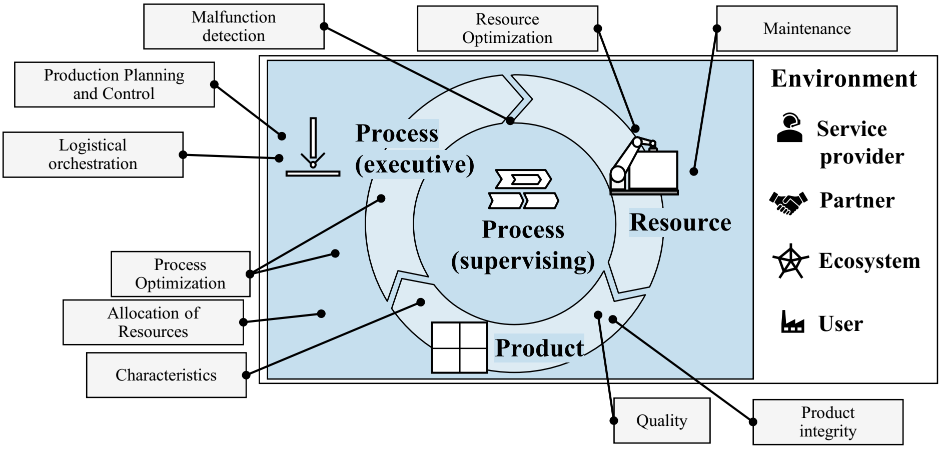


Fig. 2 PPR-Model and its elements in a smart factory environment (extended and based on [20])

according to their key characteristics and reference objects (product, process, resource). A taxonomy can be built based on this approach. The main goal of combining both approaches is to bridge the gap between the domains of manufacturing and Data Analytics. The main effect is a common baseline among stakeholders, including both data scientists and those responsible for production (see Fig. 3).

A combination of both models results in a two-view-based classification of smart factory use cases. The generic description of the use case categories is a first step to help categorize use cases into a use case matrix. The categorization can be used to group algorithm classes in use cases according to their potential. To further explain the given characterization of use cases, the following real-world examples are given:

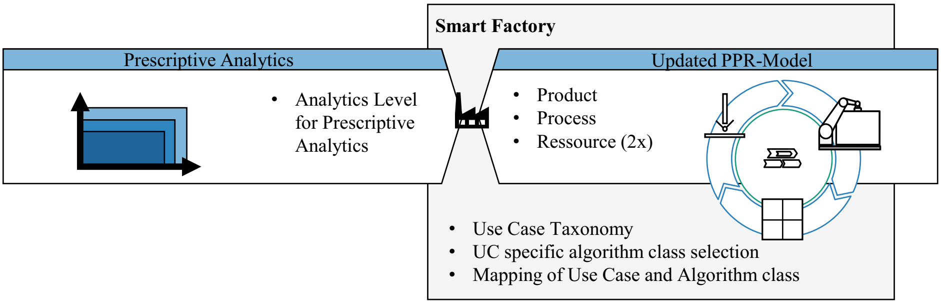

Smart Factory
Prescriptive Analytics
• Analytics Level for Prescriptive Analytics
Updated PPR-Model
• Product
• Process
• Ressource (2x)
• Use Case Taxonomy
• UC specific algorithm class selection
• Mapping of Use Case and Algorithm class

Fig. 3 Combination of Data Analytics and manufacturing domain to find a common approach for smart factory use cases

- Resource, cross-system knowledge-based reactions strategy selection: The maintenance algorithm of the scrap belt conveyor predicts a system failure in two weeks. It automatically schedules a maintenance shift and gives a recommendation how the problem can be avoided or fixed (e.g. based on a reaction strategy selection from a ontology like database).
- Process Supervising, System specific knowledge-based reaction strategy: selection: Based on changes in product demands, the production schedule is changed accordingly based on different input target values. The schedule is automatically changed.

The standardized approach to defining use cases builds the foundation for further specifying a taxonomy for smart factory use cases. They provide the basis for finding common building blocks in each use case. Efforts can be reduced heavily if the degree of reusability of the applied solution is high. The key to its success is the integration of expert knowledge and different model types. The models can be structured according to the proposed categories in the matrix.

Future work needs to focus on how well both the descriptive and prescriptive approaches to a smart factory use case work together. Based on Lepenioti et al. [12] (SLR to prescriptive Analytics Approaches) and the further implementation of use cases, one could add the dimension “algorithm” to further map which technical solution works best for which customer need (prescriptive level) based on which subdomain of the smart factory (ppr-model). A multi-dimensional space to describe use cases in (prescriptive) analytics for smart manufacturing results (see Fig. 4). The goal is to use the structuring approach to find common solution elements that can be reused or resampled. The approach generates a baseline to further reduce the need for expert knowledge during production. It frontloads the effort by implementing expert knowledge into the system. The overall goal of the approach is to reduce the investment of expert knowledge and time into the overall lifecycle of production system. The proposed approach is complementary to the approaches of Kühn et al. and Panzner et al. and can be used in a later stage of the use case development and specification.

4 Conclusion

The presented systematization scheme for (prescriptive) analytics use cases in a smart factory environment enables companies and researchers to find common attributes between different prescriptive smart factory use cases within one production ecosystem. A data-centric and a manufacturing-based view (PPR) were presented, reconfigured, and combined. The combination is derived from the Gartner analytics levels and the PPR model according to Haasis et al. [20]. The combination (P2PR and 2-dimensional characterization of prescriptive analytics) ensures a holistic view of prescriptive analytics use

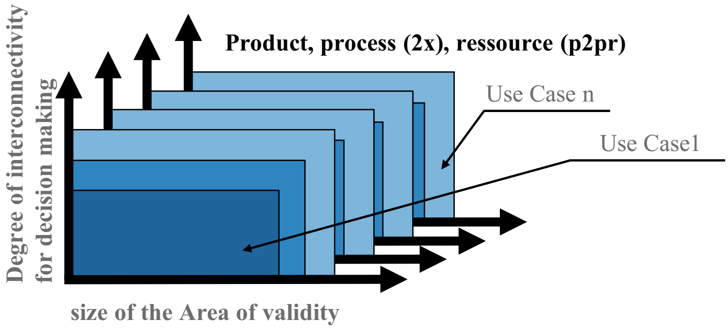

Product, process (2x), ressource (p2pr)
Use Case n
Use Case1
Degree of interconnectivity for decision making
size of the Area of validity

Fig. 4 Vision of use cases in all 4 domains, based on the two differentiation schemes

cases in manufacturing. Their combination creates a complete view from both analytics need and manufacturing prerequisites site. The resulting schema creates a first step towards a standardized taxonomy for prescriptive analytics use cases.

Further research questions arise in the context of how to combine the different prescriptive analytics approaches. Effective prescriptive analytics demands an efficient integration of expert knowledge and a high degree of reusability between use cases. The implementation of future prescriptive use cases will create a baseline to contribute to finding links between the P2PR characterization and different ML models or ML classifications of models. This approach can be embedded in the presented characterization scheme. Additionally, one needs to create a framework (technical and logical) to integrate the IT infrastructure int a shared approach to create synergies in smart manufacturing companies.

Acknowledgements This research was funded by the German Federal Ministry of Education and Research (BMBF) in the project VIP4PAPS, grant number 03VP10031. The contents of this publication are the sole responsibility of the authors.

References

1. Duan, L. Da Xu, L.: "Data Analytics in Industry 4.0: A Survey," Information systems frontiers : A journal of research and innovation, pp. 1–17, (2021) https://doi.org/10.1007/s10796-021-10190-0
2. Bianchini, M., Michalkova, V.: "Data Analytics in SMEs: Trends and Policies: OECD SME and Entrepreneurship Papers No. 15," OECD. (2019)
3. Vater, J., Harscheidt, L., Knoll, A.: "A Reference Architecture Based on Edge and Cloud Computing for Smart Manufacturing," in pp. 1–7
4. Vater, J., Harscheidt, L., Knoll, A.: "Smart Manufacturing with Prescriptive Analytics," in 2019 8th International Conference on Industrial Technology and Management (ICITM), Cambridge, United Kingdom, pp. 224–228. (2019)

5. O'Donovan, P., Leahy, K., Bruton, K., O'Sullivan, D.T.J.: "Big data in manufacturing: a systematic mapping study," J. Big Data **2**(1). (2015) https://doi.org/10.1186/s40537-015-0028-x
6. Hüllermeier, E.: "Prescriptive Machine Learning for Automated Decision Making: Challenges and Opportunities," (Dec. 2021)
7. Steenstrup, K., Sallam, R.L., Eriksen, L., Jacobson, S. F.: "Industrial Analytics Revolutionizes Big Data in the Digital Business" (2014)
8. Lade, P., Ghosh, R., Srinivasan, S.: Manufacturing Analytics and Industrial Internet of Things. IEEE Intell. Syst. **32**(3), 74–79 (2017). https://doi.org/10.1109/MIS.2017.49
9. Shao, G., Shin, S.-J., Jain, S.: "Data analytics using simulation for smart manufacturing," in *Proceedings of the Winter Simulation Conference 2014*, Savanah, GA, USA, pp. 2192–2203. (2014)
10. Dubey, R., Gunasekaran, A., Childe, S.J., Blome, C., Papadopoulos, T.: Big Data and Predictive Analytics and Manufacturing Performance: Integrating Institutional Theory, Resource-Based View and Big Data Culture. Brit J Manage **30**(2), 341–361 (2019). https://doi.org/10.1111/1467-8551.12355
11. Lee, J., Singh, J., Azamfar, M., Pandhare, V.: "Industrial AI and predictive analytics for smart manufacturing systems," in Smart Manufacturing: Elsevier, pp. 213–244. (2020)
12. Lepenioti, K., Bousdekis, A., Apostolou, D., Mentzas, G.: Prescriptive Analytics: A Survey of Approaches and Methods. In: Abramowicz, W., Paschke, A. (eds.) Lecture Notes in Business Information Processing, Business Information Systems Workshops, pp. 449–460. Springer International Publishing, Cham (2019)
13. Lepenioti, K., Bousdekis, A., Apostolou, D., Mentzas, G.: Prescriptive analytics: Literature review and research challenges. Int. J. Inf. Manage. **50**, 57–70 (2020). https://doi.org/10.1016/j.ijinfomgt.2019.04.003
14. Frazzetto, D., Nielsen, T.D., Pedersen, T.B., Šikšnys, L.: Prescriptive analytics: a survey of emerging trends and technologies. VLDB J. **28**(4), 575–595 (2019). https://doi.org/10.1007/s00778-019-00539-y
15. Vater, J., Harscheidt, L., Knoll, A.: Smart Manufacturing with Prescriptive Analytics: Proceedings of 2019 8th International Conference on Industrial Technology and Management. IEEE, Piscataway (2019)
16. Menezes, B.C., Kelly, J.D., Leal, A.G., Le Roux, G.C.: Predictive, Prescriptive and Detective Analytics for Smart Manufacturing in the Information Age. IFAC-PapersOnLine **52**(1), 568–573 (2019). https://doi.org/10.1016/j.ifacol.2019.06.123
17. Meyer, M., Panzner, M., Koldewey, C., Dumitrescu, R.: 17 Use Cases for Analyzing Use Phase Data in Product Planning of Manufacturing Companies. Procedia CIRP **107**, 1053–1058 (2022). https://doi.org/10.1016/j.procir.2022.05.107
18. Kühn, A., Joppen, R., Reinhart, F., Röltgen, D., von Enzberg, S., Dumitrescu, R.: Analytics Canvas – A Framework for the Design and Specification of Data Analytics Projects. Procedia CIRP **70**, 162–167 (2018). https://doi.org/10.1016/j.procir.2018.02.031
19. Panzner, M., Meyer, M., von Enzberg, S., Dumitrescu, R.: Business-to-Analytics Canvas - Translation of Product Planning-Related Business Use Cases into Concrete Data Analytics Tasks. Procedia CIRP **109**, 580–585 (2022). https://doi.org/10.1016/j.procir.2022.05.298
20. Haasis, S., Frank, D., Rommel, B., Weyrich, M.: Feature-basierte Integration von Produktentwicklung, Prozeßgestaltung und Ressourcenplanung. VDI Berichte, 1497th ed. (1999). Accessed: 3. Jan. 2022
21. Stanev, S.: Methodik zur produktionsorientierten Produktanalyse für die Wiederverwendung von Produktionssystemen – 2REUSE. (2012)
22. Pfrommer, J., Schleipen, M., Beyerer, J.: "PPRS: Production skills and their relation to product, process, and resource," in 2013 IEEE 18th Conference on Emerging Technologies & Factory

Automation (ETFA 2013): Cagliari, Italy, 10 - 13 September 2013, Cagliari, Italy, pp. 1–4 (2013)

23. Brecher, C. et al.: Multi-Level Modeling Framework for Machine as a Service Applications Based on Product Process Resource Models
24. Hillebrand, M.: “Entwicklungssystematik zur Integration von Eigenschaften der Selbstheilung in Intelligente Technische Systeme: Dissertation,” (2021)

Development of a Standardized Data Acquisition Prototype for Heterogeneous Sensor Environments as a Basis for ML Applications in Pultrusion

Timo Helfrich, Michael Wilhelm, Oliver Kuppler, Philipp Rosenberg, and Frank Henning

Abstract

Pultrusion of continuous fiber reinforced profiles has been state of the art for several decades. However, pultrusion in the production environment so far has no or only few sensor data in a heterogeneous sensor environment, has shortcomings in data quality (time synchronization, different formats, different sampling rates, sporadic disconnections and resulting data losses), and insufficient data processing methods are used for process control and optimization. Significant efficiency improvements would therefore still be possible in this respect. The question therefore arises as to how a data acquisition system can be designed that provides heterogeneous data in a standardized and reliable manner for data-based process development and optimization. The aim is therefore to develop a digitized and standardized data acquisition prototype for the continuous production of sustainable profile structures so flexible that the various components of the production line can be adaptively combined in a central data acquisition system. We have therefore screened and selected possible standardized transmission

T. Helfrich · O. Kuppler
Selfbits GmbH, Karlsruhe, Germany
e-mail: oliver.kuppler@selfbits.de

M. Wilhelm (✉) · F. Henning
Department of Polymer Engineering, Fraunhofer Institute for Chemical Technology (ICT), Pfinztal, Germany
e-mail: michael.wilhelm@ict.fraunhofer.de

P. Rosenberg
Pfinztal, Germany

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_10

formats in combination with suitable low-cost data acquisition systems for a highly reliable and secure application area. The paper shows a profound concept for a data acquisition prototype based on different low-cost control systems for the applicability and testability of the developed requirements regarding data acquisition, processing, and future storage.

Keywords

Pultrusion • Data acquisition • Standardized data basis

1 Introduction

With the aim of increasing the sustainability, circular economy and resource efficiency of fiber-reinforced plastics, a new process variant of the pultrusion process is being developed as part of the CaproPULL project. The so-called “in-situ pultrusion process” to produce thermoplastic profiles based on nylon 6 offers decisive advantages over the processing of non-recyclable and non-functionalizable thermoset profiles that is common in the state of the art.

To produce the thermoplastic profiles, dry fibers stored in large racks are continuously pulled through a preheating and drying unit before entering a steel die where they are impregnated with the matrix. The highly reactive two-component matrix, which is stored in a metering and mixing unit, immediately begins to polymerize in the die at defined temperatures in several heating zones. When the fully cured profile leaves the die, it cools until it reaches the puller and the saw, where it is cut to the desired length.

Due to the high number of process and material parameters of the very sensitive material, a data-based process development and optimization would significantly increase the development time and product quality and thus shorten the time to market and enable a wide use and substitution of the current non-sustainable products.

Since the pultrusion process is very simple in its principles, digitization including machine and sensor data acquisition and processing is not common today [1]. Taking into account the Industry 4.0 maturity index in Fig. 1 and the classification of [2], pultrusion may currently be assigned to the “Industry 3.0 stage” (step 1–2: computerization - connectivity), as there are predominantly no or only a few connections between the main systems pultrusion line, injection unit, preheating unit and various sensor systems.

When working with limited process data without a central unit, the synchronization of different files with different and sometimes missing timestamps, different sampling rates and multiple data sets with the same information leads to unstructured datasets and makes an accurate and time-dependent interpretation of process-induced effects time-consuming and complex, so that many effects and findings are not determined. Consequently, there is an urgent need to work out the basics and steps to prospectively implement ML methods to improve the quality of the process and profiles.

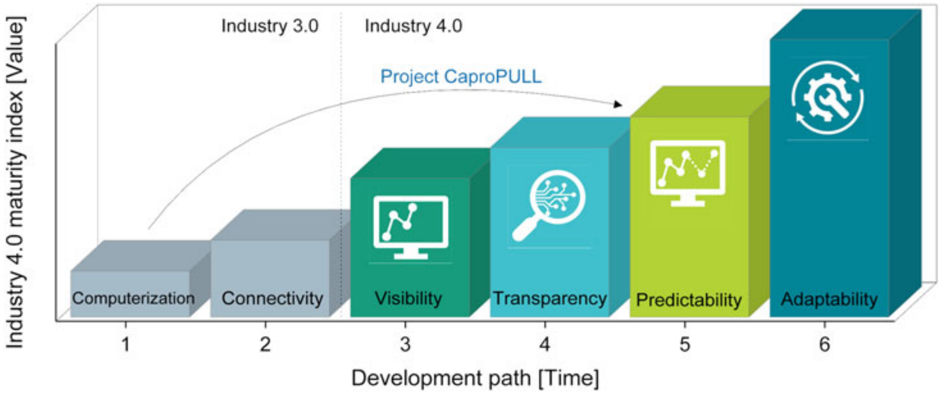


Fig. 1 Industry 4.0 maturity index and objective in Capro-PULL project. Fig. adapted from [2]

The question therefore arises as to how a data acquisition system can be designed that provides heterogeneous data in a standardized and reliable manner for data-based process development and optimization. The aim is therefore to develop a digitized and standardized data acquisition prototype for the continuous production of sustainable profile structures so flexible that the various components of the production line can be adaptively combined in a central data acquisition system.

The paper discusses the state of the art in industrial communications, highlighting standardized communication protocols that will be considered in the subsequent concept development of the standardized data acquisition prototype. This includes defined requirements for data acquisition, processing, and future storage of the given systems. Necessary modifications, extensions, and the use of retrofits of the given systems are presented based on a standard protocol, followed by a final concept evaluation.

2 Industrial Communication – State of the Art

Choice of sensors as well as transmission options and protocols heavily determine the capabilities of the data acquisition process. While the sensor technology is mostly determined by the specific experimental conditions, transmission protocols are often determined by the manufacturers of the sensor technology or the data acquisition systems. This is mainly due to the historical development of industrial communication.

Starting with parallel cabling of sensors, actuators and controllers, industrial communication has been developed since the 1980s. Initially, serial interfaces such as MODBUS RTU or PROFIBUS enabled to set up networks for exchanging information in production while maintaining real-time capability. However, the limits were the number of network participants and the communication in the IT systems [3].

The development of Industrial Ethernet intended to overcome these limits and enable communication across multiple company levels. In this context, new industrial transmission protocols and fieldbus successors such as PROFINET or EtherCAT emerged.

One of the biggest challenges of the fieldbus is the lack of protocol compatibility among each other. High investment cost for machinery requires long usage times leading to heterogenous protocol environments being the norm. Gateways or similar individual solutions are therefore necessary for the integration of devices in cross-company applications [4]. With rising awareness for the necessity of standardization of data exchange in industrial control systems standards like OPC UA emerged, as well as extensions to network protocols like MQTT. There are many common industrial protocols, but due to the advanced standardization, this paper will focus on OPC UA and MQTT.

Message Queuing Telemetry Transport (MQTT)

MQTT is a communication protocol for machine-to-machine communication. It is considered to be particularly lightweight and easy to implement. MQTT is based on the publish-subscribe model, in which messages can be distributed from one instance (publisher) to many interested parties (subscribers), usually via middleware (the broker). Messages are written to so-called topics, which can be organized in a tree structure. Clients can either subscribe to these topics, i.e. receive messages in case of updates, or publish them, i.e. publish messages themselves. The protocol offers various security mechanisms such as encryption and authentication [5]. Simplified data modeling and standardization of this is introduced with the Sparkplug B specification, which defines topic namespaces, making it suitable for industrial data exchange in IIoT scenarios [6].

Open Platform Communications Unified Architecture (OPC UA)

OPC UA is a communication standard that defines a comprehensive data model and mappings to various communication protocols. Data transport and data modeling are largely independent of each other. Both client–server and publish-subscribe models can be used for communication, with the former currently being used more frequently. A mapping to MQTT also exists. OPC UA stands out especially due to its data modeling. While fieldbuses only transmit values, these can be extended in OPC UA with the corresponding semantics. Thus, a value can be supplemented with information about the measured variable and the unit in which it is represented [7]. Key benefits of OPC UA compared to data models developed on top of existing protocols like MQTT are the fully developed and standardized information models for specific machine and equipment types, enabling vendor independent development of applications.

3 Concept Development for Machine Data Acquisition

In order to develop a concept for the collection of data, first requirements are documented, then standards are selected and consequently applied in the concept design.

3.1 Requirements for a Standardized Data Acquisition

To develop a feasible data acquisition concept for pultrusion, key requirements for such a system must be defined. Requirements result from interviews at research level, literature research and were supported by an industry questionnaire for pultruding companies to fulfill estimated future needs and demands. Requirements regarding the raw data can be deducted from the list of parameters that drive the pultrusion process. Technical requirements regarding the data acquisition, processing and storage must be carefully tailored to the peculiarities of the pultrusion process to enable analysis in the future. Resulting requirements are listed in Table 1.

3.2 Selection of Preferred Standards

Aspects like time synchronization, adequate resolution, accuracy, acquisition rate and completeness of acquired data are heavily dependent on the chosen data model and protocol, hence their selection is essential. When weighing up the various communication standards, OPC UA and MQTT stand out above all because of their built-in security mechanisms and data modelling possibilities and their high reliability.

With the so-called sessions, OPC UA enables to resend information sent in the meantime after a connection has been lost and successfully re-established. MQTT offers Quality of Service, which can be configured in three levels. The highest level ensures that the receiver has received the information exactly once. Therefore, one of these two protocols will be used at the interface between data acquisition and processing.

Table 1 Requirements for data acquisition, processing, and storage

No	Title	Requirement
1	Sampling rates	Acquisition of 80 process parameters with more than 1 Hz
2	Transmission reliability	Disconnections occur without data loss
3	Data storage	Export possibility of data given
4	Transmission security	Data transmission security against common cyber attacks
5	Time synchronization	Time synchronization of all recorded values

OPC UA as well as MQTT provide built-in data modelling possibilities. While OPC UA standardizes a comprehensive and at the same time extensible information model, the latest MQTT Sparkplug specification B allows a simplified data model to be built using a topic structure (like a folder path). Likewise, the OPC UA specifications offer a mapping of its data model to PubSub protocols like MQTT in OPC 10,000–14.

For the data model, the information model of OPC UA is used due to the extensive modeling possibilities, the already profound standardizations, and the possibility of transmission via MQTT. The novel combination of these two standards (OPC UA over MQTT) enables a new comparison of the two standards. The project will examine which transmission path (MQTT vs. OPC UA) is the more reliable and suitable approach. Thus, the goal for data acquisition is defined based on the interface for data processing. In the next step, the existing systems and components are analyzed, and integration steps are defined.

3.3 Retrofitting a Standardized Data Acquisition System

The pultrusion plant at Fraunhofer ICT exemplarily reflects a common production as it can be found in manufacturing companies as the three main components of the system show different interfaces and degrees of digitalization. The overall system is shown in Fig. 2 and consists of a pultrusion system including mold, a mixing and metering system, and a heating chamber for pre-drying, where the former is most digitized and interconnectable whereas the latter has no digital interface at all.

We define those three sub-systems as distinct scenarios as they impose different requirements on our system that is supposed to eventually integrate data from the heterogenous sources in a uniform way.

Scenario 1, The pultrusion plant and the mixing and metering plant use PROFINET for communication. At the same time, the systems are capable of implementing standards such as OPC UA or MQTT due to their state-of-the-art control technology.

Scenario 2, in a data acquisition box, sensor signals for temperatures and pressures in the mold are currently processed with an amplifier and a transmitter. Latter has a serial interface which can be processed on a standard computer using Labview.

Scenario 3, the heating chamber for pre-drying the fibers currently has no data interface. An expansion of the sensor technology is desired since the drying process influences process stability and product quality.

Digitization of Production Line Components

According to the degrees of digitization, the production line components must be integrated into a data acquisition system via several steps. In the first step, the transmission protocols and the information transmitted in the process are standardized to be able to supply constant, homogeneous data quality for each device in the future.

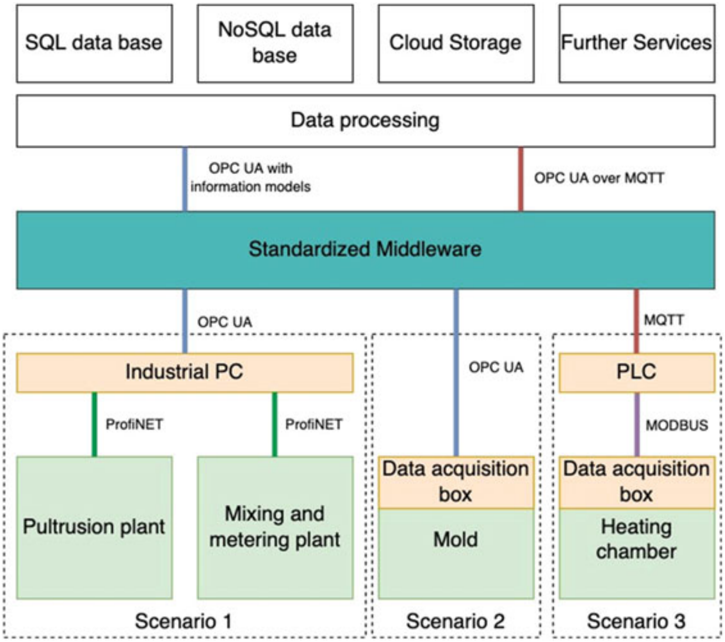


Fig. 2 Schematic overview of machine data acquisition and transmissions within the in-situ pultrusion line

Scenario 1, the pultrusion line as well as the mixing and metering unit can be upgraded to common communication standards via their PLCs. Nevertheless, it is planned to use a PROFINET-capable industrial PC as a gateway to separate the production network from the data acquisition network. Due to security considerations, they are separated with distinct hardware. The industrial PC publishes the process parameters of the two machines via OPC UA to obtain standardized access to the variables.

Scenario 2, the current Labview program can be extended with Labview's OPC UA toolkit, which allows the measurement data to be provided in an OPC UA server to the middleware. The OPC UA interface replaces the Windows solution and secures it with common security mechanisms.

Scenario 3, in a retrofit an additional data acquisition box will be installed at the heating chamber, for implementation of additional sensors (humidity and temperature). The acquisition system also consists of a transmitter and amplifiers for the individual sensors. Depending on the transmitter there are various scenarios for data recording. A transmitter with serial MODBUS interface is selected to minimize costs. This requires preprocessing of a signal to transform to common interfaces (such as OPC UA or MQTT). This is realized via programmable logic controllers (PLCs). In this project, the aim is to compare common, powerful PLCs with compact Arduino-based PLCs.

Standardized Middleware

After these adjustments the production line components can supply process data to other systems. In this state, the data is only partially standardized and synchronized in time. In order to meet the requirements and to provide data in a standardized format, a middleware is implemented as the center of data acquisition. It abstracts from the proprietary or semi-standardized protocols and represents a generic approach to data provision. It is implemented on the edge to pick up data from the components, synchronize it in time via NTP and provide it in an aggregated data model according to common OPC UA standards. So the task is to combine data and prepare it in a standardized way. It is still to be tested if data will be transmitted through OPC UA or MQTT using the OPC UA data model. This provides standardized and easy access not only for the following data processing but also to other services developed in future.

Data Processing and Data Persistence

Signal data or variables of the protocols are stream processed. They trigger a data stream, which can be modified and further processed. The standardized information models of the OPC UA and MQTT interfaces simplify the integration of data acquisition into data processing and allow the reuse of processing mechanisms. The challenge is to find out at which level such data processing can take place - directly on a PLC, on an edge device within production or in a cloud environment.

After processing, the data shall be stored permanently for documentation. For this purpose, two approaches are compared: The standardized formats of OPC UA allow the use of NoSQL databases that enable fast storage of the data. At the same time, the data model based on OPC UA also offers a basis to model a schema for a relational database that stores the data for later analysis. Parallel use is therefore possible, and the two approaches are to be evaluated in direct comparison.

3.4 Concept Evaluation

Table 2 takes up the requirements defined in Sect. 3.1 and considers their fulfillment by the developed concept.

4 Summary and Outlook

A concept for data acquisition, processing, and storage for the in-situ pultrusion process was developed as part of the CaproPULL project. Data acquisition is based on and at the same time uses novel combinations of state-of-the-art communication standards. In future trial campaigns, the developed concept will be validated in the existing production environment. For this purpose, hardware and software adaptations are currently being carried

Table 2 Evaluation of the concept against the given requirements

No	Title	Evaluation
1	Sampling rates	MQTT and OPC UA allow sufficiently high sampling rates
2	Transmission reliability	Both OPC UA sessions and MQTT QoS provide protection against disconnections and data loss
3	Data storage	Persistent data storage in NoSQL and in relational SQL database
4	Transmission security	OPC UA and MQTT offer built-in security mechanisms, Network segmentation prevents attacks
5	Time synchronization	Time synchronization by local NTP server

out. This includes the digitization of the various components, the setup of the database as well as its database modeling based on common OPC UA standards. Within the scope of the work, recommendations for actions to control the process will be derived. In addition, the development of a uniform data model provides the basis for the development of future ML and AI applications.

This work not only serves to optimize the in-situ pultrusion process, but also offers the potential to optimize already established process variants with other material combinations (e.g., thermoset matrices) in terms of economic efficiency and process robustness. This is possible as the high number of parameters and their known, but also partly unknown interactions can be structured and efficiently processed by statistical methods.

Funding The project is funded by the Ministry of Economics, Labor and Tourism in Baden-Württemberg ("CaproPULL" - grant number BW1_0025/01).

References

1. Wilhelm, F., Wedel, A., Wilfert, J., Holland, M.: Process digitalization of existing lines. WT WERKSTATTSTECHNIK, 154–160 (2022). https://doi.org/10.37544/1436-4980-2021-3-52.
2. Forschungsinstitut für Rationalisierung (FIR) e. V. an der RWTH Aachen, Ed., "Zukunft 4.0 – Digitalisierung, erneuerbare Energien & Data-Analytics," 2, 2016.
3. Wollschlaeger, M., Sauter, T., Jasperneite, J.: The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. IEEE Ind. Electron. Mag. **11**(1), 17–27 (2017). https://doi.org/10.1109/MIE.2017.2649104.
4. Galloway, B., Hancke, G.P.: Introduction to industrial control networks. IEEE Commun. Surv. Tutorials **15**(2), 860–880 (2013). https://doi.org/10.1109/SURV.2012.071812.00124.
5. MQTT Version 5.0. https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html. Accessed: 10. Mar. 2023.
6. Hochanadel, K.: Eclipse Foundation, Sparkplug MQTT Topic & Payload Specification Rev 2.2, 2019. Accessed: 10. Mar. 2023. https://www.eclipse.org/tahu/spec/Sparkplug%20Topic%20Namespace%20and%20State%20ManagementV2.2-with%20appendix%20B%20format%20-%20Eclipse.pdf.

7. Mahnke, W., Leitner, S.-H., Damm, M.: OPC Unified Architecture. Berlin, Heidelberg: Springer, Berlin (2009). http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1592687.

A Digital Twin Design for Conveyor Belts Predictive Maintenance

Marina Meireles Pereira Mafia, Naeem Ayoub, Lennart Trumpler and Jesper Puggaard de Oliveira Hansen

Abstract

Artificial intelligence has been widely used to enable predictive maintenance. However, AI systems need a large amount of data to generate accurate results that can be used reliably in terms of data quality. One of the ways to obtain data from the system is through the development of a digital twin. Therefore, a digital twin design might be of key value for the predictive maintenance of systems enabling the simulation of the system's performance, anticipating potential malfunctions, and consequently reducing the cost of unforeseen failures of the physical system. In this paper, we present a framework of a digital twin system for a conveyor belt along with different sensors that collect various types of data to be analyzed by a digital system. This way, the digital twin can generate more data focusing on reducing the time to obtain enough data to train the AI algorithm properly. Furthermore, the digital twin model is designed to develop the simulation environment and integrate it with the physical system.

M. M. P. Mafia (✉) · N. Ayoub · L. Trumpler · J. P. de O. Hansen
Technology Entrepreneurship and Innovation, TEI, University of Southern Denmark, Sonderborg, Denmark
e-mail: marinapereira@iti.sdu.dk

N. Ayoub
e-mail: naeemayoub@iti.sdu.dk

L. Trumpler
e-mail: petru20@iti.sdu.dk

J. P. de O. Hansen
e-mail: jesperp@iti.sdu.dk

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_11

Keywords

Predictive maintenance • Digital twin • Artificial intelligence • Data quality

1 Introduction

Artificial intelligence (AI) has gained much space in the industrial environment by providing increasingly accurate analysis of a large amount of data. Furthermore, with the evolution of Industry 4.0, in which it is possible to implement, collect information and analyze various sensors more quickly, artificial intelligence is favorably viewed as it enables the application of methods that generate more knowledge about the production system.

One of these applications has been in the maintenance of equipment and systems through predictive maintenance. Predictive maintenance (PM) is conceptualized as "estimations that identify potential machine breakdown, allowing for the setback source to be eliminated or maintained" [13]. With this, we can identify the use of AI as an enabler for developing and applying predictive maintenance in the industry. Pagano [11] applied Long Short-Term Memory Neural Networks and Bayesian inference in the heavy industry. Ayvaz and Alpay [2] applied predictive maintenance in a baby diapers assembly line, and for that purpose tested the Random Forest, XGBoost, Gradient Boosting, AdaBoost, Multilayer Perceptron (MLP) Regressor, Neural Network and Support Vector Regression (SVR).

However, to obtain accurate results that support predictive maintenance analyses and decisions, artificial intelligence needs a large amount of data to train the algorithms and thus obtain reliable predictions [8]. Thus, the accuracy and reliability of AI results are compromised when it comes to companies that do not have this large amount of data. Another point that some companies are currently facing regarding data accuracy is that in addition to a large amount of data, these must have a good sampling of real data. Thus, to remedy the lack of good sampling and the absence of a large amount of data, the development of a digital Twin can enable the understanding of the entire system by collecting data from the physical system and simulating various scenarios in the virtual system.

A digital twin (DT), as defined by NASA [1] is: "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin" [15].

As the term got adopted more widely in different industries beyond the aerospace sector, different viewpoints emerged. The understanding of the term started to shift depending on the industry and its specific application. However, the idea of having a bi-directional connection between a physical asset and its virtual counterpart is a common vision for the implementation of a digital twin [14]. As a promising key tool for the success of digital transformation, DT is being used in industries to improve the manufacturing process's operation and maintenance [6]. However, companies and the literature are still discussing how to structure and implement the digital twin for predictive maintenance [4]. Haghshenas et al. [7] applied DT to offshore wind farms, but the simulation developed in Unity was used to simulate the as-is scenarios and not to generate what-if scenarios to provide more data for the AI. With that, the AI used only historical data to perform the forecast, in addition, they applied the forecast to the sensors individually, not looking at the system as a whole.

This paper aims to propose a framework for how the development of a digital twin for predictive maintenance in an industrial conveyor system can be structured. The framework is part of a bigger project that involves the development of an AI-driven software framework that merges and utilizes heterogeneous technologies to identify a better trade-off between performance and energy consumption. In this way, we present and discuss how the framework was developed and the beginning of the implementation of this digital twin framework for a conveyor belt predictive maintenance system since this work is still in progress.

2 Related Work

Despite the few publications regarding the use of DT for conveyor belt predictive maintenance, some papers can be highlighted. Răileanu et al. [12] developed an architecture of an embedded DT for a shop-floor material integrated with the manufacturing, planning, scheduling, and control architecture where the DT monitors and forecasts the conveyor's operating parameters like pallet traveling time. The focus was on predicting failures based on operational data collected from the conveyor system such as transportation time, current pallet, batch identification number, and timestamp, and not by sensors measuring the performance of each conveyor component such as conveyor speed, engine, conveyor tension, among others.

Bondon et al. [3] also presented an application of the learning and identifying phases of a LIVE Digital Twin for a roller conveyor system consisting of a motor, frame, belt, rollers, and two sensors. However, as presented by the authors, data analytics and the maintenance plan were still missing. Despite having presented a methodology to design a model-based Digital Twin for prognostic and diagnostic based on four stages of analysis, this methodology did not address the issue of how information flow was implemented, what software and hardware requirements, and especially regarding the connection of equipment and sensors in the Digital Twin system.

Mahmoodian et al. [10] present a novel DT architecture design for an intelligent civil infrastructure maintenance system to monitor an offshore jetty conveyor of a port terminal. The authors also highlight the need to monitor an offshore jetty conveyor of a port terminal and include simulating maintenance measures focusing on time and cost savings in asset management. In this architecture, the virtual twin is used to simulate, based on historical performance data, the current and future structural infra-structural condition, and the what-if scenarios are used after analyzing the calibration model just to evaluate the maintenance measures alternatives. Virtual Twin in this case was not used to generate data, through what-if scenarios, to improve the artificial intelligence models' accuracy but was used as a tool to choose the best scenario for decision-making.

In this way, studies are being developed to use the Digital Twin to improve predictive maintenance, but there is still room for improvements and applications, as it is possible to identify a gap in the literature regarding the proposition of developing a DT framework for conveyor belts predictive maintenance in which it addresses the software, hardware, and communication requirements that evaluate the conveyor system components, as well as a way to connect both the Physical Twin and the Virtual Twin to a complete Digital Twin system. In addition, there is a lack of studies that use DT as a tool for generating new data about the system.

3 Framework

To identify how the DT for predictive maintenance of a conveyor system can be structured, this paper was based on the 5C architecture proposed by Lee et al. [9] and adopted by van Dinter, Tekinerdogan, and Catal [5], in which it presents the development of a DT-based to predictive maintenance based on 5 features, namely Connection, Conversion, Cyber, Cognition, and Configuration.

Connection is related to the Physical twin development, the Conversion deal with the system security, the Cyber consists of the digital twin development, the Cognition layer consists of the visualization feature and the Configuration is concerned with machine optimization. Thus, to develop the framework was structured into three topics, namely Data Connectivity and collection, PLC, and Sensors, and Virtual Twin, to ensure the feasibility of development and future implementation of this system. Subsequently, they were combined into this single framework presented in Fig. 1. Then, each topic is addressed in the following subsections.

3.1 Data Flow

Fig. 1 is divided into two parts, with the bottom rectangle representing the physical system and the top part standing for the cloud. The physical twin consists of the conveyor and the IO-Link sensors, as well as the AC motor and a linear actuator that can be controlled over Profinet and Serial communication respectively. Data will be collected from the IO-Link sensors and the frequency converter used for the AC motor. The HMI will serve as the user interface for the operator to interact with the equipment.

The gathered data is combined in a CSV (Comma-separated values) file and sent to the Cloud from the PLC using node-red. Here it is stored in a cloud-computing platform like Microsoft Azure and used to run the digital twin simulation. The saved data from the physical twin, as well as the newly generated data from the digital twin, is now made available to

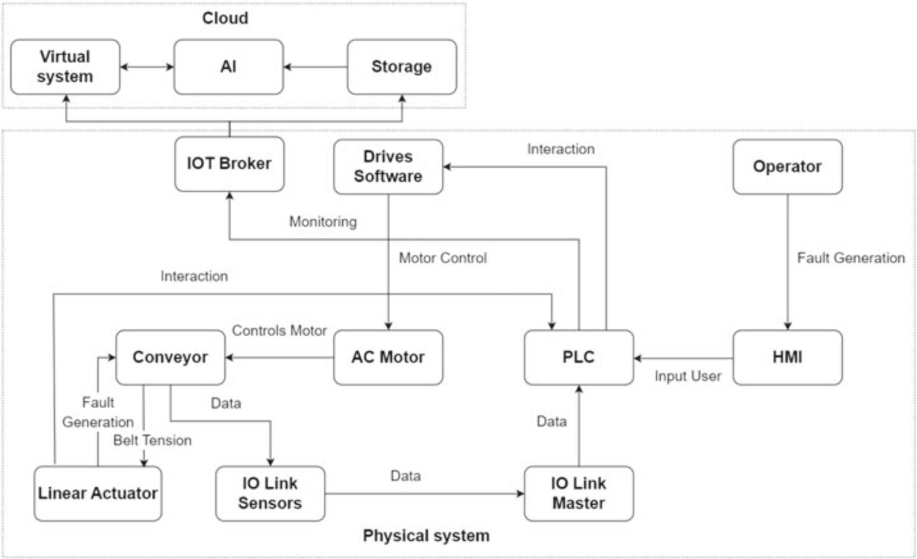


Fig. 1 Conveyor belts predictive maintenance framework

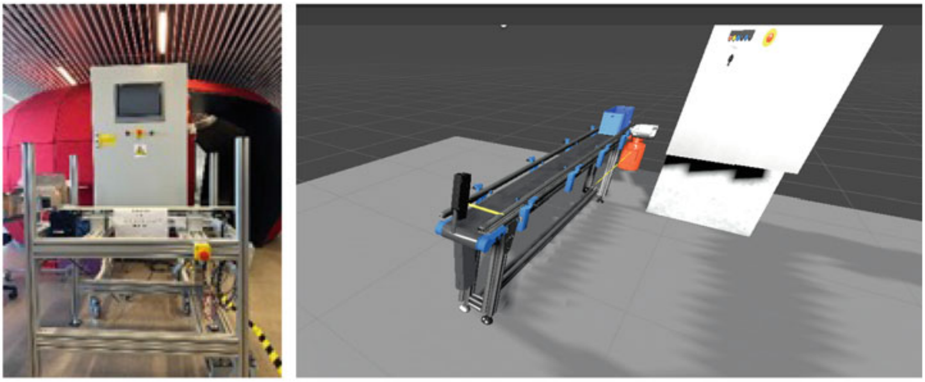


Fig. 2 Physical Twin (left picture) and Virtual Twin model (right picture)

the AI, increasing the overall amount of available data for the training process Subsequent paragraphs, however, are indented.

3.2 PLC and Sensors—Physical Twin

In the below written, we refer to the physical twin as [5]; being the physical system and the components herein. The predictive machine setup consists of a conveyor in a 1:2 gearing configuration driven by a 750-W 3-phase electrical AC motor, as depicted in Fig. 1 and Fig. 2. To this configuration, several sensors were added based on the parameters that influence the systems. These include:

- A humidity and temperature sensor for the motor, bearings, and environment
- An encoder for measuring the RPMs
- A distance sensor for measuring the position of the items on the belt
- A microphone to detect abnormal noises from the belt and motor
- A vibration sensor to detect abnormal vibrations
- A linear actuator to measure and apply force onto the belt
- A variable frequency drive to log speed, voltage, current, frequency
- A camera to perform image processing (object detection) and object placement

These sensor modules are connected to a Siemens ET200SP PLC, except for the camera, which is connected to a Xilinx board to preprocess the data before sending it into the cloud. The Siemens ET200SP was chosen as it consists of four-core CPUs in which one is dedicated to the machine operations while the three other CPUs operate the Windows operating system. This configuration makes it possible to create a PLC server that can be connected to, e.g., a client PC. The communication to and from the PLC happens through ethernet by the protocol Profinet or TCP/IP. This allows communicating quickly and safely with both Siemens and third-party equipment.

3.3 Data Connectivity and Collection—Cyber-Physical System

This section presents the connectivity and data transfer system consisting of LANs and wireless communication through TCP messages to send all sensor data to the cloud. The PLC is connected through LAN to the wireless hub, where the communication protocol is developed to transfer the data.

Continuous transfer of the data to the cloud is important to keep system memory usage lower and the performance efficiency better of the PLC. The TCP protocol is used because there are fewer chances of losing any part of data as our data. There is a possibility to also have redundant data while generating the data from sensors. To avoid sending redundant data, the preprocessing method is used to read the encoded data obtained from sensors and decode it to save the clean data in a CSV file for each day having 288 records (interval of 5 min) and then send it to the cloud server for further processing. Saving the data to CSV format help to deal with any network disturbance. In the preprocessing step, the data collected from all other sensors is directly read in PLC instead of the camera sensor. For running the object detection algorithm on the camera sensor, we used Xilinx board, which runs the real-time deep learning object detection model with 20 FPS (Frames per second).

The transferred data is then used in digital twin to further assess each sensor's performance and use it for machine learning models on the cloud-based servers.

3.4 Virtual Twin

Following the 5C framework, this section deals with the development of the digital twin. The virtual counterpart of the physical conveyor belt is executed as a digital shadow, with data flow only from the physical to the digital model [1], and the ex-tend model to a fully integrated bi-directional digital will be implemented with the AI output. The virtual twin is used to mirror the physical conveyor as well as generate new data using a simulation approach as presented in Fig. 2. Therefore, the model helps acquire large amounts of data for predictive maintenance purposes by reducing the necessary time for the physical model data collection. As the accuracy of the predictive maintenance algorithm is highly dependent on the simulated data from the digital twin, the accuracy of the virtual model plays a fundamental part in the success of the predictions. Therefore, it is necessary to simulate the entire conveyor system to explore the correlations between its components.

Based on the model parameters and the incoming sensor data from the physical twin, the digital twin simulates failures/ non-failures in the system. The simulation outputs an array of sensor data, along with the label of "failure/ non-failure" within the system, indicating whether the current parameters lead to a fault in the conveyor. This data can then be processed further for predictive maintenance purposes.

The digital twin will be implemented as an actual multi-domain model that is described by mathematical equations. The conveyor will be modeled using OpenModelica as a modeling language and using predefined components to describe the system and the relationships between its individual parts. After modeling the entire system, the input variables can be configured as needed using the data gathered from the physical twin. Based on this data the digital twin can then extrapolate new data by solving the equations describing the model. This approach will let the digital twin be able to answer questions like What will happen to the conveyor belt temperature if the belt is running at a high speed with low tension?

4 Discussion and Future Work

This paper presents a framework for developing a digital twin for generating data based on a simulation model using the collected data from a physical conveyor system. The newly generated data will then be used as input for a predictive maintenance Machine Learning algorithm to improve the model's accuracy.

For future work, we suggest using the model predictions to automatically adjust the physical system parameters, therefore closing the loop for a bi-directional digital twin model.

Acknowledgements The authors gratefully acknowledge the financial support from the ITEA3, Denmark, for the "EFICAS—Energy Efficient Heterogeneous AI-Framework for Smart Mobile and Embedded Systems" project.

References

1. Allen, B.D.: Digital twins and living models at nasa. In: Digital Twin Summit (2021)
2. Ayvaz, S., Alpay, K.: Predictive maintenance system for production lines in manufacturing: A machine learning approach using iot data in real-time. Expert Syst. Appl. **173**, 114598 (2021)
3. Bondoc, A.E., Tayefeh, M., Barari, A.: Live digital twin: Developing a sensor network to monitor the health of belt conveyor system. IFAC-PapersOnLine **55**(19), 49–54 (2022)
4. van Dinter, R., Tekinerdogan, B., Catal, C.: Predictive maintenance using digital twins: A systematic literature review. Inf. Softw. Technol. 107008 (2022)
5. van Dinter, R., Tekinerdogan, B., Catal, C.: Reference architecture for digital twin-based predictive maintenance systems. Comput. Ind. Eng. **177**, 109099 (2023)
6. Elbazi, N., Mabrouki, M., Chebak, A., Hammouch, F.: Digital twin architecture for mining industry: Case study of a stacker machine in an experimental open-pit mine. In: 2022 4th Global Power, Energy and Communication Conference (GPECOM), pp. 232–237. IEEE (2022)
7. Haghshenas, A., Hasan, A., Osen, O., Mikalsen, E.T.: Predictive digital twin for offshore wind farms. Energy Inform. **6**(1), 1–26 (2023)
8. Hoffmann, M.W., Drath, R., Ganz, C.: Proposal for requirements on industrial ai solutions. In: Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2020, pp. 63–72. Springer, Berlin (2021)
9. Lee, J., Bagheri, B., Kao, H.A.: A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manuf. Lett. **3**, 18–23 (2015)
10. Mahmoodian, M., Shahrivar, F., Setunge, S., Mazaheri, S.: Development of digital twin for intelligent maintenance of civil infrastructure. Sustainability **14**(14), 8664 (2022)
11. Pagano, D.: A predictive maintenance model using long short-term memory neural networks and bayesian inference. Decis. Anal. J. **6**, 100174 (2023)
12. Răileanu, S., Borangiu, T., Ivănescu, N., Morariu, O., Anton, F.: Integrating the digital twin of a shop floor conveyor in the manufacturing control system. In: Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future: Proceedings of SOHOMA 2019 9, S. 134–145. Springer (2020)
13. Sahli, A., Evans, R., Manohar, A.: Predictive maintenance in industry 4.0: Current themes. Procedia CIRP **104**, 1948–1953 (2021)
14. Schleich, B., Anwer, N., Mathieu, L., Wartzack, S.: Shaping the digital twin for design and production engineering. CIRP Ann. **66**(1), 41–144 (2017)
15. Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., Wang, L.: Modeling, simulation, information technology & processing roadmap. Nat. Aeronaut. Space Adm. **32**(2012), 1–38 (2012)

Augmenting Explainable Data-Driven Models in Energy Systems: A Python Framework for Feature Engineering

Sandra Wilfling

Abstract

Data-driven modeling is an approach in energy systems modeling that has been gaining popularity. In data-driven modeling, machine learning methods such as linear regression, neural networks or decision-tree based methods are applied. While these methods do not require domain knowledge, they are sensitive to data quality. Therefore, improving data quality in a dataset is beneficial for creating machine learning-based models. The improvement of data quality can be implemented through preprocessing methods. A selected type of preprocessing is feature engineering, which focuses on evaluating and improving the quality of certain features inside the dataset. Feature engineering includes methods such as feature creation, feature expansion, or feature selection. In this work, a Python framework containing different feature engineering methods is presented. This framework contains different methods for feature creation, expansion and selection; in addition, methods for transforming or filtering data are implemented. The implementation of the framework is based on the Python library *scikit-learn*. The framework is demonstrated on a use case from energy demand prediction. A data-driven model is created including selected feature engineering methods. The results show an improvement in prediction accuracy through the engineered features.

Keywords

Energy systems modeling • Data-driven modeling • Feature engineering

S. Wilfling (✉)
Institute of Software Technology, Graz University of Technology, Graz, Austria
e-mail: sandra.wilfling@tugraz.at

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_12

1 Introduction

Modeling and simulation is an crucial step in the design and optimization of energy systems. While traditional modeling methods rely on system parameters, a recent approach focuses on creating data-driven models based on measurement data taken from a system. Mainly, data-driven models are based on machine learning (ML) methods. ML methods can be classified into white-box and black-box ML methods based on their explainability [15]. For instance, Molnar et al. [15] classify methods such as linear or logistic regression and decision trees as white-box ML methods, and methods such as neural networks or decision tree ensembles as black-box ML methods.

A white-box ML model provides information about the underlying system for instance through its input-output relations (interpretability) or through a humanly comprehensible structure (explainability) [14]. To keep the structure of the model comprehensible, explainable models focus on a reduced complexity. As a result, their capability of modeling complex dependencies is often limited, creating a trade-off between accuracy and explainability [2].

To capture more complex dependencies using white-box ML models, methods of feature engineering are applied. The main purpose of feature engineering is to augment the existing dataset through adding new information, or expanding or reducing the feature set. In addition, the quality of a single feature can be improved, for instance through transformation or filtering [9].

The area of feature engineering covers a wide number of methods, such as feature creation, feature expansion [5] or feature selection [3]. Feature creation includes encodings of time-based features, such as cyclic features [19], or categorical encoding [11]. Similarly, feature expansion is the method of creating new features based on existing features. Feature expansion covers classical methods such as polynomial expansion [5] or spline interpolation [6]. In contrast to feature creation and expansion, *feature selection* aims to reduce the size of the feature set. While large feature sets may contain more information, high-dimensional feature sets may be subject to sparsity or multicollinearity. To address this, methods such as Principal Component Analysis (PCA) [10] reduce the feature set through transformation, while feature selection methods discard features. Feature selection can be implemented for instance through sequential methods, such as forward or backward selection or through correlation criteria [3]. including measures based on the Pearson Correlation Coefficient, as well as entropy-based criteria [1]. For correlation criteria, feature selection is mainly implemented through a threshold-based selection.

Mainly, the methods of feature engineering are applied during the first steps of creating a data-driven model, creating an engineered dataset to use for training [4]. However, feature engineering methods can also be used in combination with model selection procedures, such as grid search [1]. Feature engineering methods are widely used in applications from the energy domain, such as in prediction for building energy demand [20] or photovoltaic power prediction [4].

1.1 Main Contribution

To apply different feature engineering methods to the creation of data-driven models, a Python framework implementing differernt feature engineering methods was developed. The feature engineering framework is implemented in Python based on the *scikit-learn* framework and can be imported as a Python package. Compared with existing frameworks, the implemented framework focuses on providing a standardized function interface to allow the creation of different combined workflows with low effort. The functionality of the framework is demonstrated on a case study of an energy demand prediction use case. For this use case, a multi-step workflow consisting of a combination of feature engineering methods is created. The results of the case study show an improvement in prediction accuracy through the applied feature engineering workflow.

2 Method

The presented framework implements various feature engineering methods in Python based on the framework *scikit-learn*. The framework implements methods for feature creation and expansion, feature selection, as well as transformation and filtering operations. The source code of the framework is openly available on https://github.com/tug-cps/featureengineering.

Feature Creation and Expansion In the framework, different methods for feature creation and expansion are implemented. These methods create new features from time values or from expansion of existing features. To create new features, the implemented framework supports categorical encoding and cyclic encoding of time-based values.

- **Cyclic Features**: Cyclic features can be used to model time values through periodic functions [19]. In the implementation, sinusoidal signals x_{sin}, x_{cos} with a selected frequency f can be created based on a sample series n:

$$x_{sin}[n] = sin(2\pi f n) \tag{1}$$

$$x_{cos}[n] = cos(2\pi f n) \tag{2}$$

  The implementation offers the creation of features with a zero-order hold function for a certain time period, for instance $T_S = 1\, day$ for a signal with a time period of $T = 1\, week$.
- **Categorical Features**: Categorical encoding creates a representation of discrete numerical values through a number of features with boolean values [11]. In this implementation, for a number of categorical features $x_{0,....,N}$ for a feature x with discrete possible values $v_{0,....,N}$, a single feature x_i is defined as:

$$x_i = \begin{cases} 1 & x = v_i \\ 0 & else \end{cases} \tag{3}$$

The framework offers categorical encoding for time-based values as well as a division factor to create an encoding of a downsampled version of the time values.

- **Time-based Features**: The framework implements a method of dynamic timeseries unrolling to create features $x_{n-1}, x_{n-2}, ... x_{n-N}$ from an existing feature x. The method of dynamic timeseries unrolling is based on the research in [7]. In this implementation, dynamic timeseries unrolling is implemented through filter operations from the *scipy.signal* library. The dynamic features are created through the convolution of the signal x with a Kronecker delta for $i = 1...N$:

$$x_{dyn,i}[n] = x[n] * \delta[n-i] \tag{4}$$

This operation creates delayed signals $x_{dyn,1}, ..., x_{dyn,N}$. In this implementation, zero padding is used for the samples in the delayed signals, for which no values are available.

Feature Selection The framework offers several threshold-based feature selection methods, which analyze the input and target features based on a certain criterion, and then discard features with a low value of the criterion. A widely used criterion is the Pearson Correlation Coefficient, which is used to detect linear correlations between features [4]. The Pearson Correlation Coefficient calculates the correlation between two features for samples $x_{0,....,N}$, $y_{0,...,N}$ with mean values $\bar{x}$ and $\bar{y}$:

$$r_{x,y} = \frac{\sum_{i=0}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N}(x_i - \bar{x})^2 \sum_{i=0}^{N}(y_i - \bar{y})^2}} \tag{5}$$

In addition to the Pearson Correlation Coefficient, the framework provides thresholds based on non-linear dependency detection coefficients such as Maximum Information Coefficient (MIC) [17].

Transformation and Filtering Operations To transform features, the framework implements the *Box-cox* transformation as well as the square root and inverse transformation. In addition, the framework provides filtering operations, which were applied in timeseries prediction for instance in [9]. Discrete-time based filters can be implemented in Python through the functions implemented in *scipy.signal*. The *scipy.signal* library offers functions for calculating the coefficients for different types of digital filters. A digital filter of order N can be defined through the transfer function $H(z)$ in a direct form:

$$H(z) = \frac{\sum_{i=0}^{N} b_i z^i}{\sum_{i=0}^{N} a_i z^i} \tag{6}$$

The filter coefficients a_i and b_i define the behavior of the filter. The framework implements topologies such as the Butterworth [9] or Chebyshev filter. In addition, an envelope detection filter was implemented for demodulation of modulated signals. The direct form filter classes of the framework offer a simple option for extension. Different architectures can be implemented by re-defining the implemented method for coefficient calculation. This allows to create filters with different Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) structures.

3 Case Study

The framework was demonstrated on a use case from prediction for energy systems modeling. For this purpose, a mixed office-campus building was selected. A white-box prediction model should be trained based on existing measurement data provided by [18]. In the prediction of energy demand, different factors must be considered. Such factors include thermal characteristics and Heating, Ventilation, Air Conditioning and Cooling (HVAC) system behavior [13]. Additionally, building energy demand may be dependent on occupancy [8] or subject to seasonal trends [19]. Many of these factors show non-linear or dynamic behavior, which makes it difficult to address them through a purely linear model. Through feature engineering methods, these factors should be incorporated into the data-driven model.

Data-driven Model For the selected application, a data-driven model of the building energy demand should be created. To demonstrate the effect of feature engineering, two models were trained based on the existing measurement data: a basic regression model and a regression model with engineered features. The energy demand was measured during a period from 05/2019 to 03/2020, with a sampling time of $1h$ [18]. The feature set consisted of temperature data as well as data of the registrations for lectures inside the building. In addition, time-based data such as day-night changes and public or university holidays were included. For the energy demand prediction, a linear regression architecture was selected due to its simplicity and explainability as a white-box ML model. Dynamic system behavior as well as seasonality of the underlying system should be incorporated through feature engineering. Finally, the implemented models were compared to a baseline neural network model.

The implemented workflow consisted of a combination of cyclic [19] and categorical features [16], which were used to model seasonal trends, as well as of data smoothing [9] and dynamic timeseries unrolling [12]. Finally, feature selection using the Pearson Correlation Coefficient was applied, similar to the method applied by Chen et al. [4]. An overview of the implemented workflow is depicted in Fig. 1.

The model training was performed with a train-test split of 0.8 and 5-fold cross-validation. For the model with engineered features, the parameters for the steps timeseries unrolling and feature selection were determined through a grid search based on the metrics Coefficient

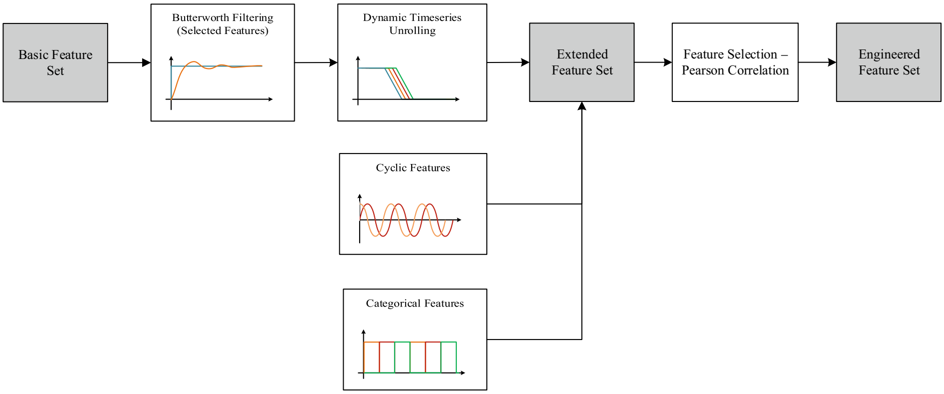

Basic Feature Set
Butterworth Filtering (Selected Features)
Dynamic Timeseries Unrolling
Extended Feature Set
Feature Selection – Pearson Correlation
Engineered Feature Set
Cyclic Features
Categorical Features

Fig. 1 Implemented workflow

of Determination (R^2), mean squared error (MSE) and Mean Absolute Percentage Error (MAPE).

Experimental Results The two models were trained on the measurement data and compared in terms of performance metrics R^2, Coefficient of Variation of the Root Mean Square Error (CV-RMSE) and MAPE. Table 1 gives an overview of the performance metrics for the different models.

The performance metrics showed a significant improvement in prediction accuracy for the linear regression model through the engineered features. This observation could also be made from the timeseries analysis depicted in Fig. 2.

The timeseries analysis showed improvements in the linear regression model especially in the seasonal trends, such as day-night or weekly behavior of the energy demand. This improvement was attributed to the introduced features. While the categorical features provided information about daily or weekly trends, the day-night behavior was modeled through the cyclic features. The introduction of the delayed input features through dynamic timeseries unrolling provided information about short-time changes in the energy demand.

The introduction of the additional features modeling the seasonality and dynamics of the energy demand showed a significant accuracy improvement for the linear regression

Table 1 Performance metrics

Model	R^2	CV-RMSE	MAPE
Linear Regression	0.316	0.328	29.176%
Neural Network	0.431	0.299	27.829%
Linear Regression – Engineered Features	0.694	0.220	17.754%

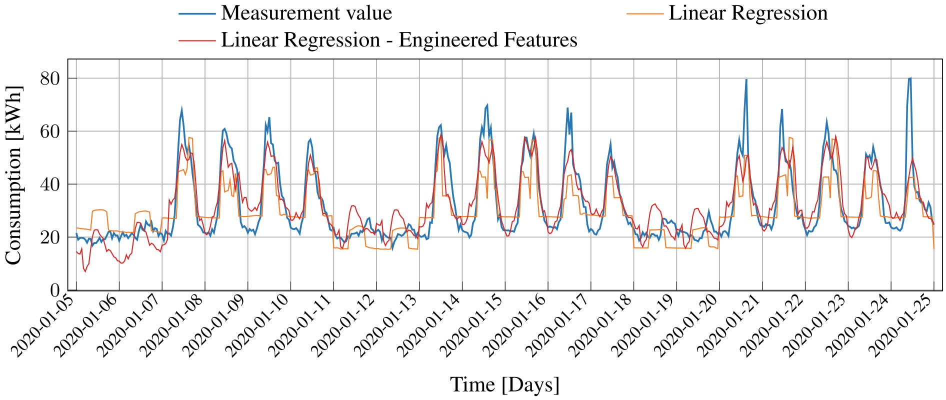


Fig. 2 Timeseries analysis for linear regression model period of 25 d from test set

model. The results suggest that this approach shows promise for improving the accuracy of explainable linear models and could furthermore be applied to non-linear methods such as neural networks or decision trees.

4 Conclusion

This work presents a Python framework for feature engineering that provides different methods through a standardized interface. The framework is based on the *scikit-learn* Python package and offers classic feature engineering methods such as feature expansion, feature creation, feature selection or transformation and filter operations. The framework is implemented as a Python package. Through the defined interfaces of the framework, additional methods can be added with low effort. Finally, the framework is demonstrated on a case study of energy demand prediction, using a workflow created from a subset of the implemented methods for data-driven model creation.

Future Work The current version of the framework gives many options for extensions. For instance, additional feature engineering methods can be added using the provided interfaces of the framework. In addition, combinations of the implemented feature engineering methods such as the implemented workflow can be used for prediction in different use cases.

References

1. Akay, M.F.: Support vector machines combined with feature selection for breast cancer diagnosis. Expert Syst. Appl. **36**(2), 3240–3247 (2009)

2. Arrieta et al.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. en. Inf. Fusion **58**, 82–115 (2020)
3. Cai, J. et al.: Feature selection in machine learning: A new perspective. Neurocomputing **300**, 70–79 (2018)
4. Chen, H., Chang, X.: Photovoltaic power prediction of LSTM model based on pearson feature selection. In: Energy Reports. 2021 International Conference on Energy Engineering and Power Systems 7 (Nov. 2021), pp. 1047–1054
5. Cheng, X. et al. Polynomial regression as an alternative to neural nets. In: arXiv:1806.06850 [cs, stat] (2019)
6. Eilers, P.H.C., Marx, B.D.: Flexible smoothing with B-splines and penalties. Stat. Sci. **11**(2) (1996)
7. Falay, B. et al.: Coupling physical and machine learning models: Case study of a single-family house. In: Modelica Conferences, pp. 335–341 (2021)
8. Ghofrani, A., Nazemi, S.D., Jafari, M.A.: Prediction of building indoor temperature response in variable air volume systems. en. J. Build. Perform. Simul. **13**(1), 34–47 (2020)
9. Gómez, V.: The use of butterworth filters for trend and cycle estimation in economic time series. J. Bus. Econ. Stat. **19**(3), 365–373 (2001)
10. Gupta, V., Mittal, M.: Respiratory signal analysis using PCA, FFT and ARTFA. In: 2016 International Conference on Electrical Power and Energy Systems (ICEPES), pp. 221–225 (2016)
11. Hancock, J.T., Khoshgoftaar, T.M.: Survey on categorical data for neural networks. J. Big Data **7**(1), 28 (2020)
12. Kumar, U., Jain, V.K.: Time series models (Grey-Markov, Grey Model with rolling mechanism and singular spectrum analysis) to forecast energy consumption in India. en. Energy **35**(4), 1709–1716 (2010)
13. Maccarini, A. et al.: Development of a Modelica-based simplified building model for district energy simulations. J. Phys. Conf. Ser. **2042**(1), 012078 (2021)
14. Manfren, M., James, P.A.B., Tronchin, L.: Datadriven building energy modelling – An analysis of the potential for generalisation through interpretable machine learning. en. Renew. Sustain. Energy Rev. **167**, 112686 (2022)
15. Molnar, C., Casalicchio, G., Bischl, B.: Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges (2020)
16. Potdar, K., Taher, S., Chinmay, D.: A comparative study of categorical variable encoding techniques for neural network classifiers. en. Int. J. Comput. Appl. **175**(4), 7–9 (2017)
17. Reshef, Y.A. et al.: Measuring dependence powerfully and equitably. J. Mach. Learn. Res. 63 (2016)
18. Schranz, T. et al.: Energy prediction under changed demand conditions: Robust machine learning models and input feature combinations. In: Proceedings of the 17th International Conference of the International Building Performance Simulation Association (Building Simulation 2021) (2021)
19. Zhang, G. et al.: Accurate forecasting of building energy consumption via a novel ensembled deep learning method considering the cyclic feature. Energy **201**, 117531 (2020)
20. Zheng, A., Casari, A.: Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists (1. edn.). O'Reilly Media, Inc. (2018)

Correction to: Towards a Systematic Approach for Prescriptive Analytics Use Cases in Smart Factories

Julian Weller, Nico Migenda, Rui Liu, Arthur Wegel,
Sebastian von Enzberg, Martin Kohlhase, Wolfram Schenck,
and Roman Dumitrescu

Correction to:
Chapter 9 in: O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems,* Technologien für die intelligente Automation,
https://doi.org/10.1007/978-3-031-47062-2_9

The affiliation of the authors Roman Dumitrescu and Wolfram Schenck was unfortunately published with an error. The initially published version has now been corrected.

The updated version of this chapter can be found at
https://doi.org/10.1007/978-3-031-47062-2_9.

O. Niggemann et al. (eds.), *Machine Learning for Cyber-Physical Systems*, Technologien
für die intelligente Automation 18,
https://doi.org/10.1007/978-3-031-47062-2_13