A MODEL LEARNING PERSPECTIVE ON THE COMPLEXITY OF CYBER-PHYSICAL SYSTEMS

Nemanja Hranisavljevic[1], **Tom Westermann**[1], **Swantje Plambeck**[2], **Henrik Sebastian Steude**[1], **Gesa Benndorf**[3], and **Oliver Niggemann**[1]
[1]Institute of Automation Technology, Helmut Schmidt University, Hamburg, Germany
[2]Institute of Embedded Systems, Hamburg University of Technology, Hamburg, Germany
[3]Fraunhofer Center for Machine Learning, Fraunhofer IOSB-INA, Lemgo, Germany

ABSTRACT

A large palette of models and their corresponding learning algorithms have been applied to time series observed from cyber-physical systems (CPSs). For some use cases, simple linear methods are sufficient, while for others, even sophisticated machine learning approaches fail to extract subtle patterns in system behavior. To date, the literature has not examined this phenomenon adequately and lacks a comprehensive analysis linking the characteristics of CPSs with the suitability of different models and learning algorithms.

In this work, after examining the complexity of multiple real-world and artificial CPS use cases, we identify several key aspects that distinguish them: 1) the number of system variables, 2) the degree of interdependence between discrete-event part and continuous part of the system, and 3) the number of unobserved system inputs. By analyzing the approaches successfully applied in the respective use cases, we were able to distill preferred techniques for addressing systems of different complexity levels.

Keywords Cyber-Physical Systems · Machine Learning · Model Learning · System Complexity · Data Dimensionality · Hybrid Dynamical Systems · Hybrid Automata

1 Introduction

Cyber-physical systems (CPSs) [19, 26] provide access to sensor measurements (such as speed or power consumption) and other variables (for example, set-point values in a control program) within a large number of devices. Together, these variables carry information about the holistic dynamic behavior of the system. This opens the door to various services based on artificial intelligence (AI), such as self-diagnosis [7] or self-reconfiguration [2], which typically rely on mathematical models of the dynamic behavior of the system. Machine learning (ML) can be used to create such models from historical data, such as system communications or sensor readings, even without much expert knowledge of the system.

The model learning and analysis of the system behavior during an actual operation phase is crucial for typical CPSs. However, at design time not all the information about the later operation is available or it is costly to gather. In such situations, the goal of model learning is to create a model of the dynamics of the system using the previously observed system data that represent training data in the ML methodology [4]. Then, given a series of successive system observations, the learned models can infer the probability of observing the given data, or predict future observations.

It is clear that many various models and approaches have been proposed to this day, from simple ones (such as nearest-neighbor-based [9]) to complex and highly parameterized ones (such as deep transformer neural networks [6]). However, the literature is not sufficiently transparent on the key aspects of the use cases studied, that could help us assess the suitability and performance of other model-learning approaches.

To address this gap, we propose the following research questions (RQs).

***(RQ 1)* From an ML perspective, how do the CPS model learning use cases differ from each other? What are simple and what are complex problems in this context?**

After investigating eight use cases from the literature, we adopt the well-established hybrid automata formalism [1, 20] as suitable for representing various CPSs. Using this representation, we propose a faceted classification scheme based on three crucial complexity aspects for the practitioner of ML (see Figure 1).[1]

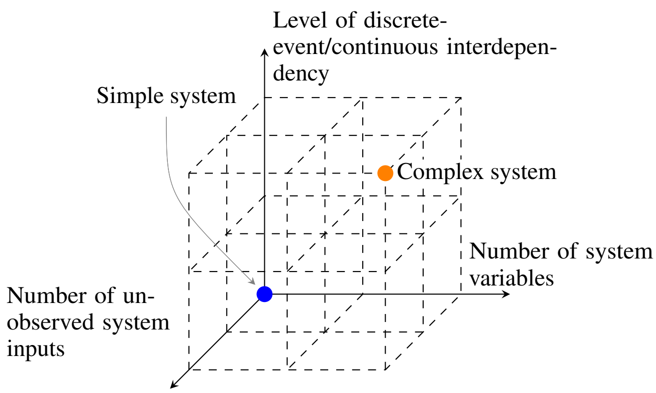

Figure 1: The three key dimensions of CPS system complexity.

***(RQ 2)* What state-of-the-art approaches are suitable for simple and what for more complex problems with respect to the three dimensions?**

We recognize ten relevant approaches from the literature that have been used effectively in anomaly detection, condition monitoring, and supervisory control. These approaches are applied to datasets of diverse degrees of complexity, according to the previously defined classification scheme. This allowed us to connect the characteristics of the methods used with the characteristics of the modeled system, as well as to explain the additional requirements and efforts required by the complex systems.

There is already a broad literature investigating the specifics of ML approaches applied in different application domains as well as in the CPS domain. For example, [15] provides an overview of model learning in industrial anomaly detection use cases. Similarly, [4] discusses the very general trends and requirements for ML in CPSs. Moreover, [18] evaluates residual-based anomaly detection approaches (autoencoder and input-output regression) on artificial CPS data. [12] compares several approaches in a manner similar to our work; however, it does not relate the facets of the system to the performance of different models. More recently, there has been work on creating scalable simulation models of different complexity (e.g. [8, 31]). In contrast, our work investigates the influence of multiple system facets on the choice of model learning techniques primarily in real-world CPS use cases.

The structure of the paper is the following. In Section 2, three aspects of system complexity are defined. Section 3 describes the selected state-of-the-art approaches. Section 4 discusses the reasons for (non-)suitability of these approaches to systems of different complexity. The conclusion and future work are given in Section 5.

[1] We limit ourselves to the three easily assessable dimensions, while other possibly relevant aspects, such as the order of system dynamics, application of the model or implemented control technique are not considered in this work.

2 CPS complexity dimensions

In order to answer *(RQ 1)*, we investigate multiple real-world and artificial use cases from the state-of-the-art literature. The use cases are selected to represent various CPS systems from different domains, including manufacturing, energy production, and space, as well as different tasks such as diagnosis or control.

1. Wind Power Plant (WPP) [10]
2. High-Rack Storage System (HRSS) [16]
3. Secure Water Treatment (SWaT) [13]
4. eBZ Assembly Plant (eBZ) [17]
5. Environmental Control and Life Support System of ISS (ECLSS) [23, 27]
6. PWM DC-DC buck converter (BuckConv) [3]
7. Excitable Cells (EC) [14]
8. Three Tank System (Tanks) [5]

In order to represent a CPS system, it is common to use the hybrid automata framework [1, 20], which is also adopted in this work (see Figure 2). The evolution of the continuous part is given by differential equations (*Diff. Eq.*), while the discrete-event dynamics is given by a transition model defining 1) switching of discrete system modes and 2) possible abrupt jumps in continuous variables. Switches are often associated with events (e_1 and e_2 in the figure), which can be triggered externally or follow from the continuous variables (e.g. when some condition turns satisfied). In addition, there is one externally triggered input e_1, and one continuous input $u_1(t)$ that affects the mode q_3.

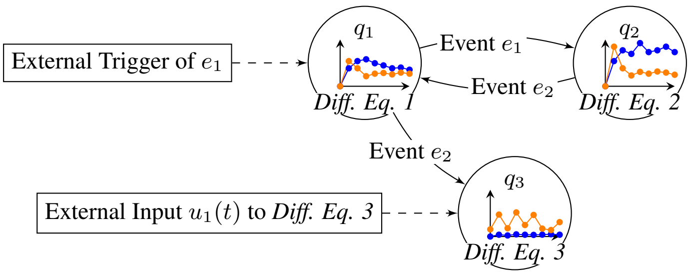

Figure 2: An illustration of a hybrid automaton with three modes (discrete states), three possible transitions and two continuous variables whose behavior is represented by different *Diff. Eq.*

In this work, we use the hybrid automata formalism to define the complexity aspects of CPSs. Namely, we represent CPS systems using a set of concurrent automata that can access each other's variables, trigger events, and share the same continuous part of the system. This is consistent with the distributed principles of CPS design (more information can be found in [1], as such details are outside the scope of this work).

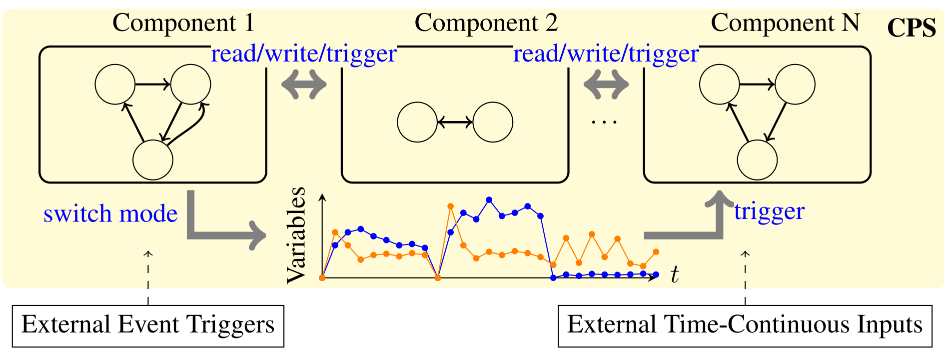

Figure 3: Set of concurrent automata as a representation of CPSs, used e.g. in [17].

2.1 Complexity Dimension 1: Number of observed system variables

It is well known that dealing with more variables of an unknown dependency structure implies a more difficult ML task, which might further lead to the so-called "curse of dimensionality" [11]. For this reason, we define the first aspect of system complexity as the number of system variables. This can be further decoupled into the number of discrete or categorical variables in the discrete event part of the system and the number of continuous variables. In the considered systems, anywhere between a few and hundreds of signals are analyzed jointly, which is undoubtedly an important parameter when choosing the model to learn. We categorize systems into three classes: less than 10 variables (class 1), 10 to 100 (class 2), and more than 100 variables (class 3) – from the simplest to the most complex, based on the information available in the literature (see Table 1). This classification takes into account solely the quantity of variables, disregarding the information they encompass, as assessing this information directly could be challenging.

System	Number of observed system variables	Class
WPP	12 continuous variables	2
HRSS	18 continuous and 3 discrete variables	2
SWaT	25 continuous and 26 discrete variables	2
eBZ	Hundreds of discrete signals published by several PLCs and the manufacturing execution system (MES)	3
ECLSS	Hundreds of continuous and discrete variables	3
BuckConv	6 continuous variables, 4 discrete variables	1
EC	7 continuous variables, 6 discrete variables	2
Tanks	10 continuous variables, 3 discrete variables	2

Table 1: Number of system variables in eight considered systems.

2.2 Complexity Dimension 2: Degree of discrete-event/continuous interdependency

In CPSs, model learning often does not consider the interactions between the discrete-event and continuous parts; then, the two parts are learned independently. Other approaches use the observed discrete data to split the continuous observations into segments corresponding to the same discrete state. However, changes in discrete signals do not always initiate mode changes in continuous behavior. This can be dealt with in different ways. For example, segments can be clustered (grouped) into those with similar behavior, or modes can be identified solely from continuous variables. Finally, the influence of continuous variables on discrete behavior is achieved by triggering a discrete state transition when conditions are satisfied. The variety of existent techniques to capture interdependency between discrete-event and continuous parts is the reason why we propose the degree of interdependency as a second complexity dimension.

It is not straightforward to evaluate the interdependency of discrete-event and continuous variables. One possible solution would be to calculate the Pearson correlation coefficient between the sequence of discrete variables and the sequence of continuous variables. Another possible solution could be to count the number of discrete (continuous) variables that are involved in the continuous (discrete) execution. In this work, we give only a rough estimate of the complexity level (1,2,3) based on the description of the system in the respective literature (see Table 2), such that:

1. if the discrete-event and continuous parts are modeled separately, we assign level 1 to the system,
2. if the influence is modeled only in one-way, we designate it as level 2, and,
3. if it was necessary to model it bidirectionally, then we assign level 3 to the system.

System	Level of discrete-event/continuous interdependency	Class
WPP	Only continuous variables are modeled in order to estimate normal functioning of the system. Clusters in the continuous behavior are only implicitly modeled.	1
HRSS	The position sensors trigger the changes of discrete modes in conveyors while each of the conveyors can be on or off causing different behavior of power, voltage and other signals.	2
SWaT	Many discrete and continuous variables influencing each other.	3
eBZ	The system has mostly discrete variables working in a timed manner and based on the temporal dependency of discrete variables.	1
ECLSS	Many discrete and continuous variables influencing each other.	3
BuckConv	Discrete events triggering state transitions depend on continuous variables	2
EC	Discrete variables are thresholded output variables.	2
Tanks	Discrete variables only enable mode. transitions	1

Table 2: Level of discrete-event/continuous interdependency in eight considered systems.

2.3 Complexity Dimension 3: Number of unobserved input variables

The systems under consideration encompass multiple, and frequently numerous, signals. Typically, the observed data variables comprise signals pertinent to system behavior and are relatively straightforward to observe, given the availability of sensors and data acquisition capabilities. However, not all relevant input signals can be observed, which can contribute to the stochastic behavior of the system, behavior that remains unexplained by the observed variables. Modeling systems characterized by significant stochastic behavior presents a more complex learning task. The categorization of the systems discussed is based on the estimated number of unobserved inputs (disturbances) that impact the system dynamics. They are classified into three distinct categories: those free from unobserved inputs (class 1), those containing fewer than five (class 2), and those comprising more than five unobserved inputs (class 3).

System	Number of unobserved system variables	Class
WPP	Discrete variable representing current gear of the system and continuous power consumption are some of the unobserved signals.	2
HRSS	Relevant discrete signals are available (HRSS v1), but not used in some experiments (HRSS v2).	3
SWaT	All relevant signals are considered to be available.	1
eBZ	All relevant signals are considered to be available.	1
ECLSS	Besides user inputs such as set-point adjustments, the system is influenced by short-term and long-term trends driven by the space station's orbit, affecting operational dynamics. Presence and other behavior of astronauts is also unobserved.	3
BuckConv	All relevant signals are available.	1
EC	Only voltages are observed. Internal currents and state variables are not observed.	2
Tanks	Only one tank level is observed while other tank levels are hidden.	2

Table 3: Number of unobserved system variables in eight considered systems.

System classification in 3 dimensions

In order to visualize the complexity of the use cases from the literature we create a 3D scatter plot (see Figure 4). Several points appear obvious:

- As expected, synthetic datasets are typically simpler than real-world ones.
- To some extent, complexity tends to grow along all dimensions simultaneously.
- There is significant diversity among the systems considered.

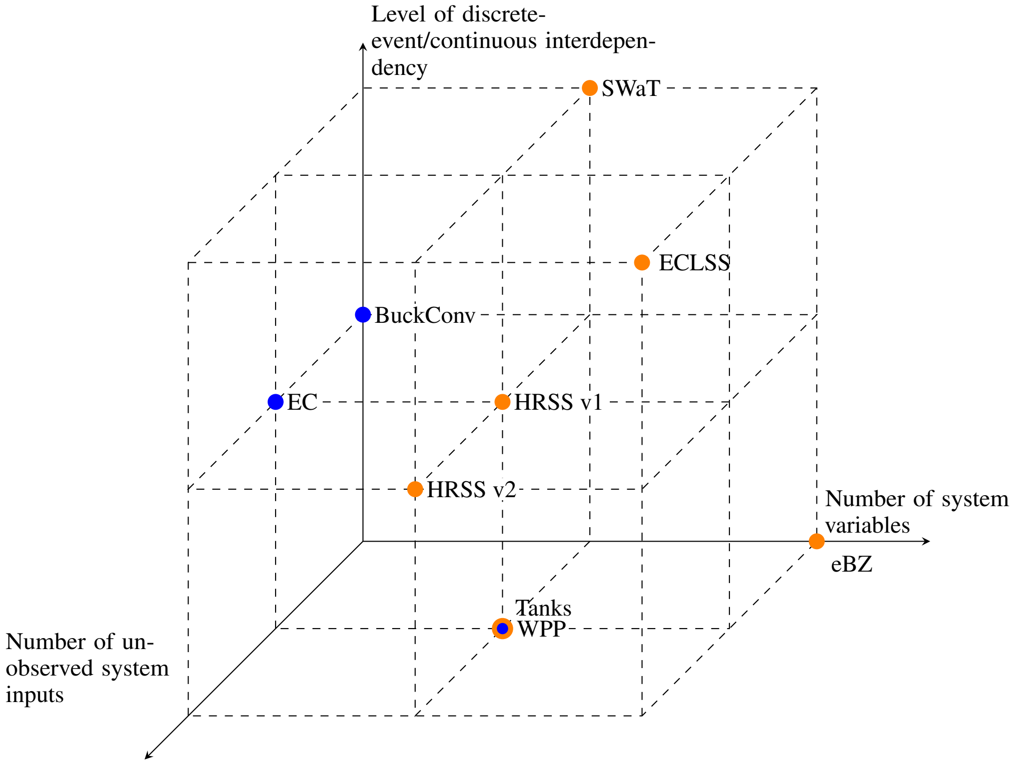

Figure 4: The three key dimensions of CPS system complexity. Blue circles represent synthetic and orange circles represent real-world use cases.

The levels of complexity dimensions are selected exactly to highlight the differences between the systems, while the complexity values of particular systems are typically not too difficult to estimate. In this way, with the proposed faceted scheme, we answer *(RQ 1)* of this work.

3 Model learning approaches

This section presents a range of model-learning approaches that have been proposed in the literature to model the respective systems in Section 2. Each approach is briefly described in terms of its methodological framework, the underlying assumptions, and specific implementation details. This overview sets the stage for a subsequent evaluation (Section 4) of their suitability across varying complexity profiles. The approaches examined, from oldest to youngest, are as follows.

- 2012 [22] In the HyBUTLA approach, switches of continuous dynamics appear as: 1) controlled switches as a consequence of a change (discrete event) in the discrete data. The continuous data are then segmented whenever a change in the discrete data appears. 2) autonomous switches, which are not manifested in the discrete data and must be identified in order to segment the continuous data into segments that belong to the same mode (ODE). This is achieved using wavelet analysis to detect switches in the continuous behavior and by clustering the model parameters for each segment. The behavior of the discrete part is learned by comparing sequences of future events for pairs of segments in a specific bottom-up order.

- 2014 [21]: OTALA (Online Timed Automaton Learning Algorithm) is an online, passive algorithm for automatically identifying timed behavior models in production systems, designed specifically for discrete data. It learns from real-time observed data without needing stored datasets or system queries. OTALA maps unique signal vectors to system states and captures timed events to model dynamic behaviors. The resulting timed automaton supports real-time diagnostics by detecting anomalies and deviations efficiently.
- 2015 [9]: A static approach is presented which combines PCA (used to reduce data dimensionality) and the nearest-neighbor method to estimate the regions of normal data points. While this method focuses on the stochastic nature of the system, it aims not in modeling system dynamics. The proposed approach is compared with DBSCAN and spectral clustering which were outperformed.
- 2019 [10]: AE A static deep autoencoder architecture is applied to capture the stochastic nature of observed data points. Similarly to some previous approaches, it lacks extension to capture dynamics.
- 2020 [16]: DENTA (DEep Network Timed Automaton) approach is based on the following: 1) The DENTA network: a stacked network of restricted Boltzmann machines to extract a latent binary representation of time series data; 2) The DENTA automaton: a transition model of the latent binary variables which is learned in the second step from the encoded binary representation. The latent automaton model incorporates the probability distribution of the next occurring event given the current state of the automaton.
- 2021 [25]: POSEHEAD is an offline learning strategy for hybrid automata. First, a sliding-window strategy is applied to segment signals. Then, similar segments are detected using dynamic time warping. Transitions are derived on the basis of changes in the inputs as well as the residence time in a state. Finally, continuous flow functions of the dynamical modes are identified with polynomial regression.
- 2022 [30]: HAutLearn is a learning algorithm for hybrid automata with four major steps. First, change points are identified in the signals. Afterwards, segments with similar solution spaces are identified using LMI. In the third step, event conditions in the form of linear inequalities are found using the random sample consensus. Finally, the modes are merged and folded into a hybrid automaton model from a prefix tree acceptor.
- 2022 [29]: In this work, a model of a dynamical system is learned as a heterogeneous Petri net using the DyClee clustering algorithm (Dynamic Clustering algorithm for tracking Evolving Environments). With DyClee, multidimensional data from a system is clustered adaptively over time. The clustering consists of a distance-based clustering in a first stage and a density-based clustering in a second stage. The density-based stage again consists of a global and a local clustering. To avoid a functionality shift over time, a forgetting process in the form of a decay function is included.
- 2023 [28]: In [28], a TCN-VAE based seq2seq model is introduced that learns discrete and continuous variables simultaneously. The model is designed to capture all unobservable variables and hidden system states during training, encoding them in the latent space of the multivariate time series data it models. This architecture employs a specialized latent space, explicitly designed to enable anomaly detection at the subsystem level. The so-called composite latent space is structured to reflect the subsystem layout of a CPS, facilitating both failure isolation and the identification of cross-subsystem anomalies.
- 2024 [24]: FaMoS learns the dynamic model of a hybrid system in four steps: trace segmentation, segment clustering, mode identification, and model construction. The segmentation and clustering are based on the similarity of signals, i.e., Euclidean distance and dynamic time warping. The continuous dynamics in the mode identification phase is characterized by differential equations. Finally, the model is represented in a decision tree, which evaluates to the currently active mode based on the previous mode and the input.

4 Interweaving system and model characteristics

In this section, we evaluate the approaches described previously with respect to the complexity of the use cases in which they were applied. Rather than evaluating the methods in isolation, the analysis focuses on how each approach performed in specific contexts, characterized by the degrees of complexity introduced in Section 2. This evaluation provides insight into the suitability of different methods based on the complexity profiles of their real-world applications, highlighting trends in their effectiveness and adaptability – which is *(RQ 2)* of this work.

Figure 5 shows the complexity levels of the system(s) to which each of the investigated approaches has been successfully applied. The following conclusions can be inferred:

- Neural network-based approaches (DENTA, Composite-Latent-TCN-VAE) are more common for more complex systems, especially for the third dimension of complexity (unobserved system inputs). The affected systems have complexity (2,2,3) and (3,3,3), where used triplets represent three dimensions of complexity.
- Interestingly, when only continuous behavior is modeled in systems with complexity (·,1,·), static approaches (neglecting sequentiality of the data) are often sufficient (AE, PCA + kNN).
- The proposed approaches based on linear systems (HAutLearn, FaMoS, POSEHEAD) are more common for simple and artificial datasets of complexity (1,2,1) or (1,2,2).
- The continuous-only use cases of complexity (2,1,2) are common and addressed by diverse techniques (AE, PCA+kNN, DyClee).

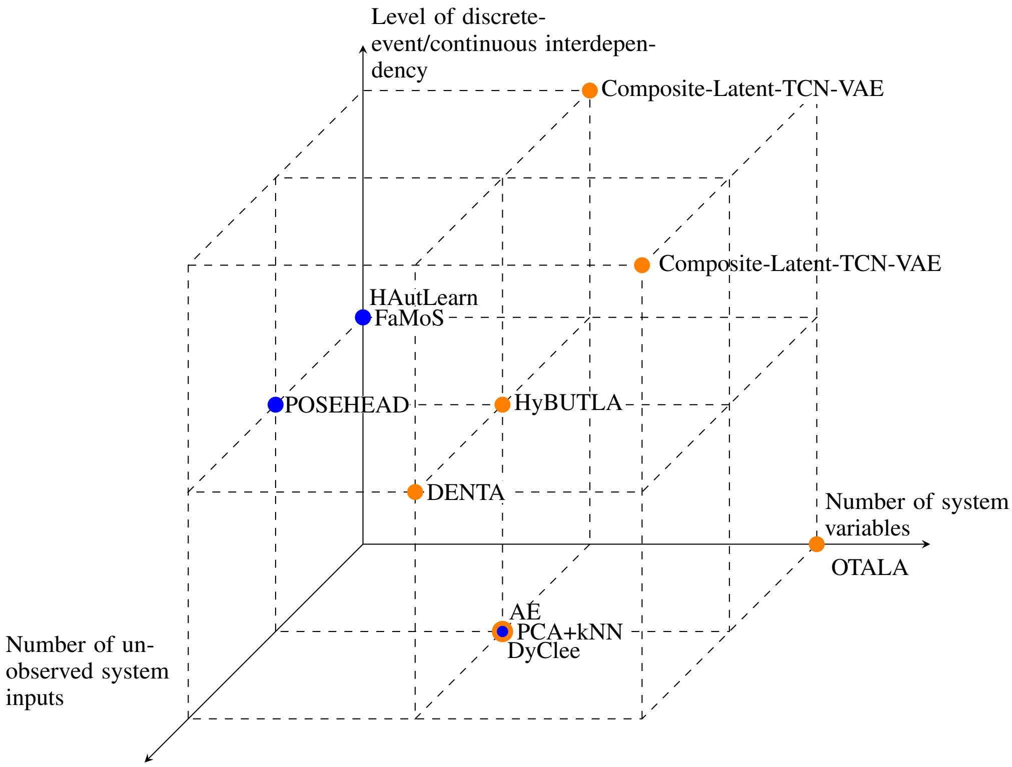

Figure 5: The state-of-the-art approaches visualized on the faceted classification scheme. Blue circles represent synthetic and orange circles represent real-world use cases.

5 Conclusion and future work

Various solutions for model learning have been proposed in the literature. However, it is hard to answer the question of their compliance with other CPS use cases. This work contributes in this direction by 1) proposing three aspects of system complexity which are relevant for model learning in CPS data; 2) analyzing methods based on their suitability to the systems of different complexity.

By interweaving system and model characteristics, this work evaluates the examined approaches with respect to the complexity of the use cases in which they were applied. Rather than evaluating the methods in isolation, the analysis focuses on how well each approach performed in specific contexts characterized by degrees of complexity. This evaluation provides insights into the suitability of different methods based on the complexity profiles of their applications.

Consequently, for a novel use case, upon categorizing a system according to its complexity, a machine learning practitioner is able to ascertain the relevance of various model-learning techniques.

Future research efforts should focus on incorporating additional use cases and exploring new methodological approaches. Furthermore, evaluating the methodologies on the same datasets would substantiate the validity of the proposed classification framework.

Acknowledgements

This research as part of the projects (K)ISS and ProMoDi is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr which we gratefully acknowledge. dtec.bw is funded by the European Union NextGenerationEU. This work was also partially developed within the Fraunhofer Cluster of Excellence ”Cognitive Internet Technologies”.

References

[1] R. Alur. Principles of Cyber-Physical Systems. Technical report, 2015.

[2] K. Balzereit and O. Niggemann. AutoConf: A New Algorithm for Reconfiguration of Cyber-Physical Production Systems. *IEEE Transactions on Industrial Informatics*, 2022. Publisher: IEEE Computer Society.

[3] O. A. Beg, H. Abbas, T. T. Johnson, and A. Davoudi. Model validation of pwm dc–dc converters. *IEEE Transactions on Industrial Electronics*, 64(9):7049–7059, 2017.

[4] J. Beyerer and O. Niggemann. Machine Learning in Automation. *At-Automatisierungstechnik*, 66(4):281–282, 2018.

[5] B. O. Bouamama, R. M. Alaoui, P. Taillibert, and M. Staroswiecki. Diagnosis of a two-tank system. Technical report, Intern Report of CHEM-project, USTL, 2001.

[6] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng. Learning Graph Structures With Transformer for Multivariate Time-Series Anomaly Detection in IoT. *IEEE Internet of Things Journal*, 9(12):9179–9189, 2022.

[7] A. Diedrich, A. Maier, and O. Niggemann. Model-based Diagnosis of Hybrid Systems using Satisfiability Modulo Theory. Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), 2019.

[8] J. Ehrhardt, M. Ramonat, R. Heesch, K. Balzereit, A. Diedrich, and O. Niggemann. An AI benchmark for Diagnosis, Reconfiguration & Planning. Sept. 2022.

[9] J. Eickmeyer, P. Li, O. Givehchi, and O. Niggemann. Data Driven Modeling for System-Level Condition Monitoring on Wind Power Plants. In *In: The 26th International Workshop on Principles of Diagnosis (DX-2015)*, 2015.

[10] B. Eiteneuer, N. Hranisavljevic, and O. Niggemann. Dimensionality Reduction and Anomaly Detection for CPPS Data using Autoencoder. In *20th IEEE International Conference on Industrial Technology*, volume 2019-Febru, Melbourn, Australia, Mar. 2019.

[11] J. H. Friedman. On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality. *Data Min. Knowl. Discov.*, 1(1):55–77, 1997. Publisher: Kluwer Academic Publishers Place: Hingham, MA, USA.

[12] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo. An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–10, Aug. 2021.

[13] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, editors, *Critical Information Infrastructures Security*, pages 88–99, Cham, 2017. Springer International Publishing.

[14] R. Grosu, S. Mitra, P. Ye, E. Entcheva, I. V. Ramakrishnan, and S. A. Smolka. Learning cycle-linear hybrid automata for excitable cells. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control*, pages 245–258, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[15] A. Happonen, U. Usmani, and J. Watada. A Review of Unsupervised Machine Learning Frameworks for Anomaly Detection in Industrial Applications. pages 158–189. July 2022.

[16] N. Hranisavljevic, A. Maier, and O. Niggemann. Discretization of hybrid CPPS data into timed automaton using restricted Boltzmann machines. *Engineering Applications of Artificial Intelligence*, 95:103826, 2020.

[17] N. Hranisavljevic, T. Westermann, P. Kroke, and C. Waschkies. Using ML-Based Models in Simulation of CPPSs: A Case Study of Smart Meter Production. In O. Niggemann, J. Beyerer, M. Krantz, and C. Kühnert, editors, *Machine Learning for Cyber-Physical Systems*, pages 19–29, Cham, 2024. Springer Nature Switzerland.

[18] C.-C. Hsu, G. Frusque, and O. Fink. *A Comparison of Residual-based Methods on Fault Detection*. Sept. 2023.

[19] E. A. Lee. Cyber Physical Systems: Design Challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369, 2008.

[20] J. Lunze and F. Lamnabhi-Lagarrigue. *Handbook of Hybrid Systems Control: Theory, Tools, Applications*. Cambridge University Press, 2009.

[21] A. Maier. 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, July 2014.

[22] O. Niggemann, B. Stein, A. Vodenčarević, A. Maier, and H. Kleine Büning. Learning Behavior Models for Hybrid Timed Systems. In *Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*, pages 1083–1090, Toronto, Ontario, Canada, 2012.

[23] P. Parodi, Z. Szigetvari, R. Muller, and A. Quaglia. Columbus eclss: five years of operations and lessons learned. In *43rd International Conference on Environmental Systems*, page 3414, 2013.

[24] S. Plambeck, A. Bracht, N. Hranisavljevic, and G. Fey. FaMoS– Fast Model Learning for Hybrid Cyber-Physical Systems using Decision Trees. In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '24, New York, NY, USA, 2024. Association for Computing Machinery. event-place: Hong Kong SAR, China.

[25] I. Saberi, F. Faghih, and F. S. Bavil. A Passive Online Technique for Learning Hybrid Automata from Input/Output Traces. *ACM Transactions on Embedded Computing Systems*, 22(1):1–24, 2021.

[26] J. Schlick. Cyber-physical systems in factory automation - Towards the 4th industrial revolution. In *Factory Communication Systems (WFCS), 2012 9th IEEE International Workshop on*, page 55, May 2012. ISSN: Pending.

[27] H. S. Steude, C. Geier, L. Moddemann, M. Creutzenberg, J. Pfeifer, S. Turk, and O. Niggemann. End-to-end MLOps integration: a case study with ISS telemetry data. In *ML4CPS – Machine Learning for Cyber-Physical Systems*. Helmut-Schmidt-Universität Hamburg, 2024.

[28] H. S. Steude, L. Moddemann, A. Diedrich, J. Ehrhardt, and O. Niggemann. Diagnosis driven Anomaly Detection for CPS. *arXiv preprint arXiv:2311.15924*, 2023.

[29] A. Vignolles, E. Chanthery, and P. Ribot. Hybrid Model Learning for System Health Monitoring. In *Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*, pages 7–14, Pafos, 2022. Science Direct.

[30] X. Yang, O. A. Beg, M. Kenigsberg, and T. T. Johnson. A Framework for Identification and Validation of Affine Hybrid Automata from Input-Output Traces. *ACM Transactions on Cyber-Physical Systems*, 6(2):1–24, 2022.

[31] B. Zimmering, O. Niggemann, C. Hasterok, E. Pfannstiel, D. Ramming, and J. Pfrommer. Generating Artificial Sensor Data for the Comparison of Unsupervised Machine Learning Methods. *Sensors*, 21:2397, 2021.