MLOps for Cyber-Physical Production Systems
Challenges and Solutions

Faubel, Leonhard; Woudsma, Thomas; Kloepper, Benjamin; Eichelberger, Holger; Buelow, Fabian; Schmid, Klaus; Ghezeljehmeidan, Amir Ghorbani; Methnani, Leila; Theodorou, Andreas; Bang, Magnus

DOI
10.1109/MS.2024.3441101

Publication date
2024

Document Version
Final published version

Published in
IEEE Software

MLOps for Cyberphysical Production Systems

Challenges and Solutions

Leonhard Faubel, University of Hildesheim, 31141 Hildesheim, Germany

Thomas Woudsma, Prodrive Technologies, 5692 EM Son, The Netherlands

Benjamin Kloepper, Capgemini Invent, 10785 Berlin, Germany

Holger Eichelberger, University of Hildesheim, 31141 Hildesheim, Germany

Fabian Buelow, ABB Corporate Research Germany, 68526 Ladenburg, Germany

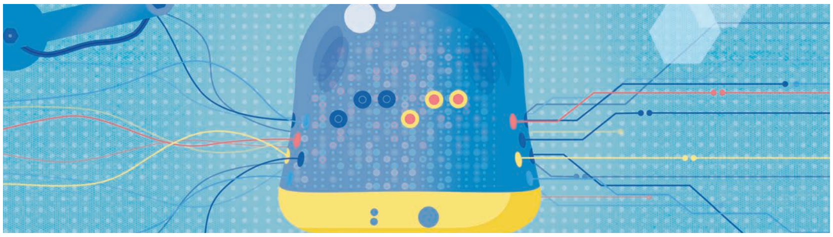
Klaus Schmid, University of Hildesheim, 31141 Hildesheim, Germany

Amir Ghorbani Ghezeljehmeidan, Delft University of Technology, 2628 CD Delft, The Netherlands

Leila Methnani, Umeå Universitet, 901 87 Umeå, Sweden

Andreas Theodorou, Universitat Politècnica de Catalunya, 08242 Manresa, Spain

Magnus Bång, Linköping Universitet, 581 83 Linköping, Sweden



Digital Object Identifier 10.1109/MS.2024.3441101
Date of publication 14 August 2024; date of current version 11 December 2024.

// *While machine learning operations (MLOps) have received significant interest, much less work has been published addressing MLOps in industrial production settings lately, particularly if solutions are not cloud based. This article addresses this shortcoming based on our and our partner's real industrial experience across different application domains.* //

MACHINE LEARNING (ML) can boost performance and effectiveness in industrial processes. However, the processes, techniques, and tools for bringing ML models into operation in the industrial environment go far beyond in-lab requirements. In production, models must deal with live data subject to frequent changes and can be influenced by external sources.[1] Lately, this also led to the introduction of machine learning operations (MLOps) techniques in the context of cyberphysical production systems (CPPS). While significant work exists on general continuous integration and delivery challenges and mitigation strategies in cyberphysical systems,[2] ML challenges in Industry 4.0,[3] MLOps challenges and solutions in general,[4] and on MLOps-challenges in CPPS,[5] so far, less work exists on specific solutions in the CPPS domain. Here, we focus on the main challenges and accompanying solutions from the background of CPPS in industrial companies. Our list of 50 challenges and 53 solutions[6] is based on various, sometimes very different, cases from CPPS. This article focuses on main challenges, such as ensuring high-quality sensor data, managing differences in data distributions, and including domain knowledge in model building. Solutions include periodic calibration, interactive user interfaces, lightweight integration, and establishing serialization formats. Deploying options such as edge devices, on-premise solutions, and standardized formats help reduce dependencies and improve monitoring. Explainability enhances trust and maintainability by providing insights into model decisions and enabling human intervention when necessary. As our discussion of challenges and solutions is specific to CPPS and based on a significant number of projects, we believe it can be highly beneficial to practitioners in the area of CPPS. The results we present here have been drawn from our observations and studies as well as from outside interviews and collaborations.

Industrial MLOps

As ML has found its way into industrial practice, the need for MLOps as tools, processes, and practices to manage the ML lifecycle of data acquisition, preprocessing, development, and operations grows. MLOps includes continuous deployment, monitoring, and retraining and requires both ML and domain experts. In this article, we discuss experiences from different scenarios in projects such as IIP-Ecosphere and EXPLAIN (https://www.iip-ecosphere.de/, https://explain-project.eu/), from the industrial cooperating partners, from the author companies and their customers. To illustrate the diversity of scenarios, we present three of them, which we can share in more detail, from the areas of electronics manufacturing, metal production, and chemical plants. Table 1 summarizes several interesting characteristics of these scenarios. However, it should be emphasized that the overall experiences that contributed are significantly larger.

Scenario 1: Electronics Manufacturing

Electronics manufacturing involves various quality inspections. Visual inspections, e.g., to check components' presence, correctness, and alignment, are carried out at inspection stations on the assembly line. Here, automated ML systems replace humans for visual inspection. Eventually, operators check the overall production quality and the detected defects again. It is crucial for the system that there are no slips or false negatives, as the products are packaged and dispatched immediately after the ML-based inspection if no defect is detected. Further, the number of false positives is minimized to avoid operators being overwhelmed with unnecessary checks. The company leverages MLOps to react quickly if the false positive or negative rate increases and, if necessary, to retrain and provide a new model as quickly as possible.

Scenario 2: Metal Production

Metal production involves several steps, e.g., separating and enriching minerals from native rock: a process known as *froth flotation*. This process involves 1) grinding the raw material into a fine powder, placing it in a bucket/cell, 2) adding water and reagents, and 3) pumping air into the mixture to create bubbles. The hydrophilic minerals attach to the air bubbles rising to the top of the cell, where they can be extracted, and the nonhydrophilic sink to the bottom, where they can also be extracted or sent to subsequent processing. ML is used to model the froth flotation process to help the operators keep an overview and support them in their decisions. Here, MLOps helps to scale and maintain a large number of models.

Table 1. Scenarios of cyberphysical production systems.

	Scenario 1	Scenario 2	Scenario 3
Company FTE	2,300	5,000	1,000
Application Domain	Visual Inspection	Process Monitoring	Process Control
AI Problem	Inspection Automation	ML-Based Soft Sensors	Process Monitoring
No. of Models (Pipeline)	20	10	50
Model Update Frequency	Few Months	User-Triggered	User-Triggered
Triggers	Manual	Manual	Manual
Type of Retraining	Cumulative Dataset	Expert Selection	Expert Selection
Type of Deployment	Kubernetes Web Services	Kubernetes Web Services	Kubernetes Web Services
Kinds of Models	Convolutional Neural Networks	Transformer Architecture	Autoencoder

FTE: full-time equivalent.

Scenario 3: Chemical Plant

A chemical plant produces chemicals continuously while maintaining the correct quality of the finished product. Such a plant aims to maximize production yield, and any downtime in the process would result in a significant financial loss. A team of plant operators is usually responsible for a specific chemical plant with thousands of sensor signals. It is not feasible for the operators to monitor the vast amount of signals simultaneously. Instead, ML methods assist operators in monitoring the plant and detecting anomalies to prevent unplanned downtime. MLOps helps to scale anomaly detection operations in chemical plants. Further, it increases reliability in using deployment strategies such as shadow mode. Additionally, it allows for quality monitoring using drift detection methods and automated ML model retraining on demand.

Approach

As the main focus was on understanding challenges that are specific to implementing MLOps for CPPS, we gathered an author team with a focus on experience in a large number of projects with MLOps aspects in CPPS. In a first stage, we gathered challenges based on the various experiences the participants were involved in as well as from previous studies on challenges with author involvement.[1,5] This led to the identification of 50 distinct challenges.[6] A selection of the scenarios underlying these challenges is presented in Table 1 to illustrate the diversity of the CPPS cases used as a basis. In a second stage, all participants explored solutions for each challenge by leveraging real-world scenarios from authors' companies, studies, or use cases from other indirect collaborators and partners. This multistep approach yielded 53 solutions to the 50 distinct challenges.[6]

We clustered the identified challenges into five main groups. These groups were defined based on both categorization used in previous overview of challenges[5] as well as on discussions among the authors, as follows:

1. data related
2. model related
3. operational
4. socio-technical
5. support.

Due to space restrictions, we focus here only on the first three categories as we believe the others are not that specific to CPPS. For the same reason, we can only discuss a selection of representative challenges next. The remaining categories (socio-technical and support) have been addressed in detail by Siegmund et al.[7] and Kreuzberger et al.[8] Other support challenges, such as security and compliance solutions, are similar to nonfactory systems and have, therefore, been omitted. Our unique contribution lies in leveraging first-hand accounts from MLOps implementations to highlight key challenges and propose practical solutions, focusing solely on applicability in CPPS situations.

Figure 1 provides an overview of MLOps, with the discussed categories (yellow) and tasks (blue) highlighted, along with the workflow illustrated by arrows.

Challenges and Solutions

Data Related

ML requires a large amount of high-quality data to work best. Hence, data-related challenges are central to the success of MLOps.

Cyberphysical systems have to deal with data collection from hardware, like cameras and other sensors (acceleration, temperature, and so on). Data quality cannot be guaranteed per se due to issues like incorrect calibration of sensors, sensor drift, physical damage, or format changes after firmware updates. To deal with such issues, data can be checked and treated (i.e., up-/downsampling, filtering, missing-value, and outlier handling) by default during ingestion. For example, major quality issues can be monitored via ML-based solutions such as autoencoders (e.g., variational autoencoders), which can detect out-of-distribution data. Furthermore, periodic sensor calibration, verification, and physical maintenance increase data quality.

Data from similar sensor types may have different distributions for the same input stimuli, even if sensors provide high-quality data. For this reason, ML models trained with data from one sensor often do not work well with data from another. For

Deploying options such as edge devices, on-premise solutions, and standardized formats help reduce dependencies and improve monitoring.

example, in Scenario 1, an image of a product may look differently if a camera gets replaced. Data annotations or separate training datasets are created for each sensor to cope with this problem. Furthermore, data aggregation, feature engineering, and selection support model generalization by forcing the model to focus not on the data distribution of a sensor but on the actual information relevant to the problem at hand. As a result, e.g., slight changes in overall brightness may not be considered a relevant feature.

Another challenge is that many sensor manufacturers define custom interfaces and data types. These interfaces are used to transfer sensor data and metadata to the ML pipelines and the data storage. For MLOps, metadata should include additional information such as location, timestamps, equipment numbers, collection conditions, or sensor settings, allowing for ML model maintenance, optimization, and downstream analysis, e.g., fault localization and traceability. Frequently used interfaces are REST interfaces and fieldbus interfaces like EtherNet/IP or ProfiBus. A more standardized solution is OPC UA,[9] an independent open source communication protocol many sensor and programmable logic controller (PLC) manufacturers support.

ML applications in the industrial domain are often used to ensure product and process quality. Gathering data from abnormal and anomalous cases that might be required for accurate model training is costly on components, equipment, and time. For instance, training automated visual inspection models requires damaged products to gather images of nonconforming cases. Using anomaly detection models that can be trained using only data from expected behavior can save on these resources as no labeling or gathering of data from rare plant states is required. Running nominal production and collecting that data suffices. Additionally, with deployment time passing, rare cases can be continuously included in the data sets to retrain models and improve their performance.

For classification tasks, data from cyberphysical systems requires domain experts to provide high-quality

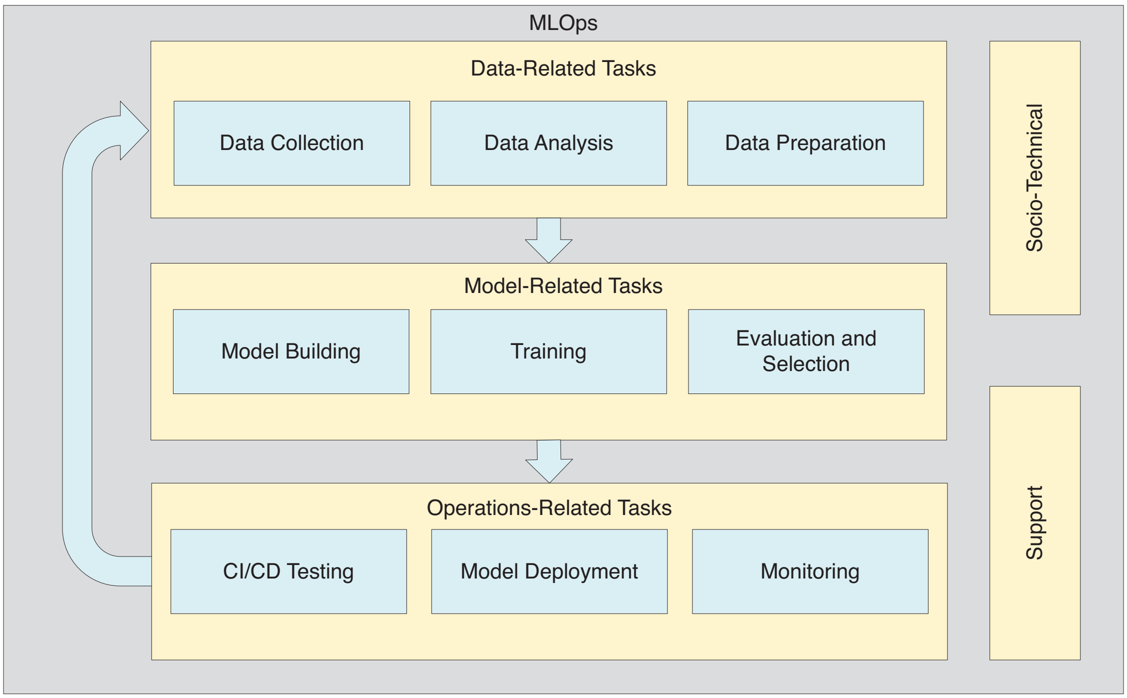


FIGURE 1. MLOps in industrial applications (adapted from Faubel et al.[5]) Only the MLOps activities in which we name and describe MLOps challenges and solutions are listed here. Other activities are omitted.

labels. This process can be costly because domain experts are typically busy running a factory. One solution is to use pretrained models with some knowledge about the data. This approach requires less experienced personnel to apply final labels. Using streamlined labeling interfaces that fit into the software environment the experts are familiar with is the preferable way to reduce cost while increasing user acceptance. Data synthesis and fusion with data from process simulation is another way of obtaining missing data for classification, regression, or forecasting tasks.

Model Related

A core element of MLOps is the model, which imposes several related challenges.

Experience shows that domain knowledge from the industrial process is essential when building and deploying machine learning models in production environments. A model that gives wrong output (classification, prediction, and so on) will be ignored or switched off by the users in the best case and can lead to production loss or unsafe states in the worst case. Domain knowledge, especially when dealing with consequences of incorrect outputs, means understanding the objectives, prediction constraints, and feature engineering based on the most relevant information for good performance, considering adequate evaluation metrics, identifying the potential bias and noise sources, and safety and legal issues. Exploiting the available domain knowledge can be simplified and streamlined by interactive UIs and tutorials.

In a typical production process environment, hundreds to thousands of sensor readings can either be used in whole or in part as features or provide new insights into the data by applying various transformations. Therefore, the processed input data must be efficiently stored and labeled for the specific applications in all scenarios.

> Data quality cannot be guaranteed per se due to issues like incorrect calibration of sensors, sensor drift, physical damage, or format changes after firmware updates.

The feature engineering and selection that support the quality and robustness of a model are particular challenges. One solution is an efficient feature storage mechanism. It contains detailed information for each application and each ML model, e.g., the data used for training. One of its benefits is the ability to use the information to optimize memory usage on edge devices.

For Scenarios 2 and 3, the model has different real-time requirements for data ingestion, model inference, and training. In such cases, different pipelines are deployed: one for data ingestion and another for model inference. Model (re-)training can be done with a third pipeline, often dependent on the two former pipelines, and is currently triggered manually. For instance, when data drift is detected in pipeline data ingestion, crucial data are missing (detected during data ingestion), or the model output is an out-of-normal distribution or of poor accuracy (detected during model ingestion). The latter case typically results from data issues. Still, it can also result from hardware or connectivity problems, shortcomings in the currently selected best model, or when model drift occurs. Then, it can be helpful to have a worse-performing but more robust (e.g., due to fewer input features) backup model that is used until a new best model is trained. Also, experience shows that it is helpful to train several models or model architectures, e.g., models for specific plant states or periods.

Production environments are typically not static, i.e., they change over time. This can be due to changing equipment, a switch in input material, or many other reasons. To account for this variability, it is important to have robust data ingestion, storage setup, model performance monitoring, and a decent number of well-performing models trained and retrained. Depending on the situation, model selection and deployment can either be executed automatically, e.g., via an accuracy metric and the recent history or via data triggers that indicate a different state or operation mode, or models can be chosen by users (operators in the plant, plant engineers, and so on). Experience shows that trust in the ML solution is typically higher if users choose the model, often in discussion with the developing data scientists. However, this leads to increased reaction time for updating models as users might not be available or constantly require changes to the model. Data on experiments, trained models, and their parameters can be stored, versioned, chosen, and used for inference using an MLOps framework such as MLflow.[10]

Typically, the initial process of data inspection, feature engineering and selection, preliminary model training, and selection happens offline, i.e., without connection to the physical system. Therefore, the execution environment is usually a data scientist's workstation or a cloud environment. Considering these processes' compute resources and environment, the choice between an in-house and an Internet cloud environment hinges on several factors. The cloud has the advantage of scalable resources that enable the handling of large datasets and complex computations; however, the availability of these resources should still be considered. Additionally, companies often have established guidelines for data security concerns with which the chosen environment should be aligned. The final decision may require a thoughtful balance between individual preferences, project requirements, resource availability, and the imperative need for data security and compliance with the organizational guidelines.

Operations Related

Finally, we address the challenges associated with operation. These are often related to the deployment of ML models, testing, and monitoring. The options for deployment to a factory environment are edge, on-premise, and external servers. Edge devices and on-premise solutions can provide short-latency inferencing. On the other hand, external servers are often used for applications that do not need to respond within a few milliseconds and for ML training:

- Edge devices are devices that collect and communicate information, e.g., PLCs and Industrial PCs (IPCs).
- On-premise solutions run on the premise of an organization. They include local servers, private clouds, and others.
- External servers are typically accessed remotely via the Internet and can be company-owned data centers or (public) cloud providers.

Edge devices and on-premise solutions can provide short-latency inferencing. On the edge, focusing on intelligent data collection or data processing and scoring of ML models can reduce resource needs.

In our experience, predictions can be made within a few 100 ms on an on-premise Kubernetes cluster. Selected tasks may also run on a PLC, but computationally intensive tasks, e.g., visual analytics, may be a factor of two to four slower. In such cases, parallelization can help to reduce mean times. However, distributing compute and ML tasks requires flexible infrastructures, allowing for distributed deployment, update, and maintenance.

External servers are not used in the three scenarios, as there are concerns regarding the provision of data. Further, in Scenario 1, the rent of a server would be more cost-intensive than using own hardware.

Further, reducing dependencies on the different technologies used in the production environment is essential and eases integration and testing. This is achieved by employing established serialization formats such as OPC UA JSON[9] for data payloads or ONNX[11] for serialization of machine learning models. Further, virtualization techniques such as KinD (Kubernetes in Docker) can reduce dependencies on heterogeneous hardware in CPPS.

During operations, various aspects must be monitored, such as ML performance and potential concept drift, resource utilization, and execution times. However, drift detection methods may be specific to the underlying data, e.g., tabular or time series data, and require experiments with actual approaches. Different monitoring forms can be integrated (e.g., with Seldon Core,[4] Bento ML,[4] or Splunk-based dashboards[12]) for metrics and monitoring model output scores.

One practical example of an open source approach, which provides many of the abovementioned solutions out of the box, is oktoflow,[13] an AI-enabled stream-based Industry 4.0 platform initially developed in the IIP-Ecosphere project. It focuses on heterogeneous on-premise hardware, lightweight integration of AI services realized in different programming languages, model-driven generation of application code, test templates, and deployment containers as well as the use of standards, in particular OPC UA and Industry 4.0 Asset Administration Shells. In oktoflow, data interfaces, as suggested previously, are mandatory for specifying models of services, ML pipelines, and applications and for ensuring their validity before code generation and deployment. Due to resource limitations of current industrial PLCs, deployment in oktoflow happens directly via virtualization such as Docker (Kubernetes support is optional). Regarding monitoring, oktoflow combines generated and application-specific probes and integrates them through Prometheus.

Explainability

Since ML is often regarded as a black box, trust, reliability, and maintainability in production and development are limited. It is especially challenging to communicate algorithms' predictions and decisions to developers or operators who do not have a background in ML. Therefore, not only the results of the ML model but also insights into how and why the model arrives at the respective results need to be communicated to them. Explainable AI (XAI) techniques provide explanations for the decisions or predictions made by a model.[14] At first, providing insights into model predictions, XAI makes it easier for MLOps development teams to interpret and validate model behavior and select the most robust and accurate model. Further, the explanations can support monitoring the ML models and improve retraining capabilities. Since enabling autonomous decisions by ML can be particularly problematic, feedback mechanisms are often used that display predictions and explanations, allow human decisions to be made based on the results, and allow intervention in case of misbehavior.

In the ITEA project EXPLAIN,[15] an end-to-end ML lifecycle was developed (Figure 2) to create an MLOps software architecture that provides explainability and interactivity for users. The architecture enables users with little ML knowledge to participate in the entire process, e.g., data preparation, modeling, model construction, and inference. We try to make the ML operations accessible and transparent, increasing the comprehensibility of the models and enabling interactions with the models to strengthen the users' domain knowledge. The implementation of such explainability and feedback mechanisms is highly context dependent.

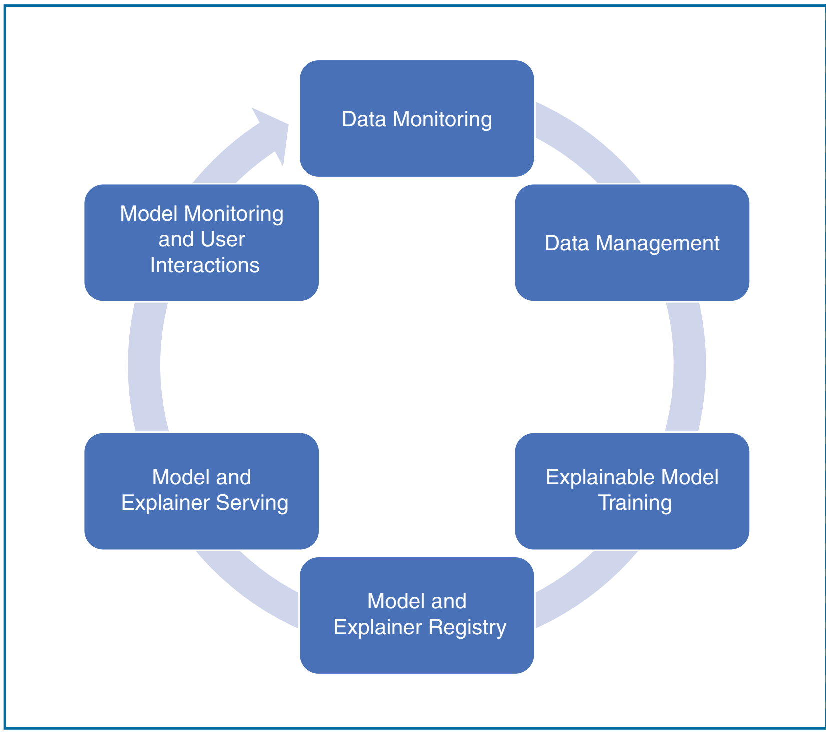


FIGURE 2. Explainable MLOps process based on the MLOps software architecture described in Faubel et al.[15]

ML has proven valuable for optimizing performance and effectiveness in industrial practice. Industrial MLOps, which involves collaboration among domain experts, ML experts, and operators, ensures ML solutions' seamless integration and functioning within cyberphysical systems. However, successfully implementing ML models in production requires data-related, model-related, operations-related, support, and socio-technical considerations.

In this article, we pointed out potential solutions to several problems in industrial MLOps. We hope this helps to create an overview of this rapidly growing field. Further, we observed that the more MLOps is used in practice, the greater the importance of explainability.

Acknowledgments

This work was supported by the project EXPLAIN, funded by the German Federal Ministry of Education under Grant 01—S22030E, funded by the Netherlands Enterprise Agency RVO under Grant AI212001, funded by Sweden's Innovation Agency (Vinnova) under Grant 2021—04336, and the project IIP-Ecosphere funded by the German Ministry of Economics and Climate Action (BMWK) under Grant 01MK20006C. Any opinions expressed herein are solely by the authors and not the funding agencies. We acknowledge financial support by Stiftung Universität Hildesheim.

References

1. K. Feichtinger et al., "Industry voices on software engineering challenges in cyber-physical

production systems engineering," in *Proc. IEEE 27th Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Stuttgart, Germany, 2022, pp. 06–09, doi: 10.1109/ETFA52439.2022.9921568.
2. F. Zampetti, D. Tamburri, S. Panichella, A. Panichella, G. Canfora, and M. Di Penta, "Continuous integration and delivery practices for cyber-physical systems: An interview-based study," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 3, pp. 1–44, 2023, doi: 10.1145/3571854.
3. R. Peres, X. Jia, J. Lee, K. Sun, A. Colombo, and J. Barata, "Industrial artificial intelligence in industry 4.0 - Systematic review, challenges and outlook," *IEEE Access*, vol. 8, pp. 220,121–220,139, 2020, doi: 10.1109/ACCESS.2020.3042874.
4. G. Symeonidis, E. Nerantzis, A. Kazakis, and G. A. Papakostas, "MLOps-definitions, tools and challenges," in *Proc. IEEE 12th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 0453–0460.
5. L. Faubel, K. Schmid, and H. Eichelberger, "MLOps challenges in industry 4.0," *SN Comput. Sci.*, vol. 4, no. 6, p. 828, 2023, doi: 10.1007/s42979-023-02282-2.
6. L. Faubel et al., 2024, "Table of MLOps challenges and solutions in CPPS based on a survey," Zenodo. [Online]. Available: https://zenodo.org/records/12725378
7. A. Mailach and N. Siegmund, "Socio-technical anti-patterns in building ML-enabled software: Insights from leaders on the forefront," in *Proc. IEEE/ACM 45th Int. Conf. Softw. Eng. (ICSE)*, Melbourne, Australia, 2023, pp. 690–702, doi: 10.1109/ICSE48619.2023.00067.

ABOUT THE AUTHORS

LEONHARD FAUBEL is a research assistant at the Software Systems Engineering Group, University of Hildesheim, 31141 Hildesheim, Germany. His research interests include software engineering and machine learning. Faubel received his M.Eng. in electrical engineering from the HAWK University of Applied Sciences and Arts. Contact him at faubel@sse.uni-hildesheim.de.

THOMAS WOUDSMA is a software architect at Prodrive Technologies, 5692 EM Son, Netherlands. His research interests include machine learning for industrial automation applications and building machine learning platforms and applications using state-of-the-art model architectures and tools. Woudsma received his M.Sc. in electrical engineering from Eindhoven University of Technology. Contact him at thomas.woudsma@prodrive-technologies.com.

BENJAMIN KLOEPPER is a senior manager at Capgemini Invent, 10785 Berlin, Germany. His research interests include the AI transformation of industrial operations with applications to chemical, automotive, and railway industries. Kloepper received his Ph.D. in AI planning for mechatronic systems from Paderborn University. Contact him at benjamin.kloepper@capgemini.com.

HOLGER EICHELBERGER is a senior Ph.D. researcher at the Software Systems Engineering Group, University of Hildesheim, 31141 Hildesheim, Germany. His research interests include variant-rich software systems, model-driven software development, (AI-) platform engineering, and software performance. Eichelberger received his Dr. rer. nat. in computer science from the University of Hildesheim. Contact him at eichelberger@sse.uni-hildesheim.de.

FABIAN BUELOW is a senior research scientist at ABB Corporate Research Germany, 68526 Ladenburg, Germany. His research interests include AI in process industry, machine learning application development, as well as modeling, simulation, and mathematical optimization of production processes. Buelow received his Ph.D. in process engineering from the Karlsruhe Institute of Technology. Contact him at fabian.buelow@de.abb.com.

ABOUT THE AUTHORS

KLAUS SCHMID is a full professor of software engineering at the University of Hildesheim, 31141 Hildesheim, Germany. His research interests include software product lines, software architectures, and adaptive/intelligent systems. Schmid received his Ph.D. in computer science from University of Kaiserslautern. Contact him at schmid@sse.uni-hildesheim.de.

AMIR GHORBANI GHEZELJEHMEIDAN is a Ph.D. candidate at the Electronic Components, Materials and Technology (ECTM) Group, Delft University of Technology, 2628 CD Delft, The Netherlands. His research interests include the reliability and failure models of electronics, with current focus on machine learning application in electronic components reliability. Ghorbani Ghezeljehmeidan received his M.Sc. in mechanical engineering from Polytechnic of Turin. Contact him at amir.ghorbani@tudelft.nl.

LEILA METHNANI is a WASP-HS affiliated Ph.D. student and a member of the Responsible AI group at Umeå University, 901 87 Umeå, Sweden. Her research interests include explainable AI in industrial applications (process industry and games) with an emphasis on human-centric explanations. Methnani received her M.Sc. in AI from Umeå University. Contact her at leila.methnani@umu.se.

ANDREAS THEODOROU is an assimilated associate professor at the Universitat Politècnica de Catalunya, 08242 Manresa, Spain. His research interests include design, implementation, and deployment of intelligent systems such as to ensure their trustworthiness. Theodorou received his Ph.D. in computer science from the University of Bath. Contact him at a.theodorou@bath.edu.

MAGNUS BÅNG is a senior associate professor and lab leader at the Computer and Information Science Department, Linköping University, 581 83 Linköping, Sweden. His research interests include human–AI collaboration and hybrid-AI decision support systems for human operators working in control rooms and includes the process industry and traffic management [ATM/UTM, train and maritime (VTS)]. Contact him at magnus.bang@liu.se.

8. D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31,866–31,879, 2023, doi: 10.1109/ACCESS.2023.3262138.
9. S. Cavalieri, M. G. Salafia, and M. S. Scroppo, "Integrating OPC UA with web technologies to enhance interoperability," *Comput. Standards Interfaces*, vol. 61, pp. 45–64, Jan. 2019, doi: 10.1016/j.csi.2018.04.004.
10. N. Hewage and D. Meedeniya, "Machine learning operations: A survey on MLOps tool support," 2202, *arXiv:2202.10169*.
11. R. S. Siddeshwara, V. Sai Rohit, P. A. Pasha, and A. S. Manakar, "Design and development of modern day machine learning applications—A survey," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 9, no. 6, pp. 251–260, Dec. 2022.
12. K. Subramanian, *Introducing the Splunk Platform. Practical Splunk Search Processing Language*. Berkeley, CA, USA: Apress, 2020.
13. H. Eichelberger and C. Niederée, "Asset administration shells, configuration, code generation: A power trio for industry 4.0 platforms, in *Proc. IEEE 28th Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Sinaia, Romania, 2023, pp. 1–8, doi: 10.1109/ETFA54631.2023.10275339.
14. A. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.
15. L. Faubel et al., "Towards an MLOps architecture for XAI in industrial applications," 2023, *arXiv:2309.12756*.