SN Computer Science (2023) 4:828
https://doi.org/10.1007/s42979-023-02282-2

ORIGINAL RESEARCH

MLOps Challenges in Industry 4.0

Leonhard Faubel[1] · **Klaus Schmid**[1] · **Holger Eichelberger**[1]

Received: 4 April 2023 / Accepted: 28 August 2023
© The Author(s) 2023

Abstract
An important part of the Industry 4.0 vision is the use of machine learning (ML) techniques to create novel capabilities and flexibility in industrial production processes. Currently, there is a strong emphasis on MLOps as an enabling collection of practices, techniques, and tools to integrate ML into industrial practice. However, while MLOps is often discussed in the context of pure software systems, Industry 4.0 systems received much less attention. So far, there is only little research focusing on MLOps for Industry 4.0. In this paper, we discuss whether MLOps in Industry 4.0 leads to significantly different challenges compared to typical Internet systems. We provide an initial analysis of MLOps approaches and identify both context-independent MLOps challenges (general challenges) as well as challenges particular to Industry 4.0 (specific challenges) and conclude that MLOps works very similarly in Industry 4.0 systems to pure software systems. This indicates that existing tools and approaches are also mostly suited for the Industry 4.0 context.

Keywords MLOps · Machine learning · Industry 4.0 · Challenges

Introduction

Industry 4.0 aims at the next industrial evolution in manufacturing, this time based on digital technologies. A core part of it is the use of machine learning (ML) to enable more intelligent, flexible, and efficient industrial production processes. Scenarios like lot-size one, predictive maintenance, or supply-chain optimization can significantly transform business models in Industry 4.0 [1]. Currently, Machine Learning Operations (MLOps) as a collection of methods, techniques, and tools for integrating ML into software development practice is widely discussed as an enabler for large-scale ML applications [2, 3]. Currently, research on MLOps focuses mostly on software systems without an embedded component.

As Industry 4.0 aims at applying ML to industrial production, the need for MLOps in this context is clear. Thus, an important question is whether the specific context of Industry 4.0, i.e., complex, large-scale Cyber-Physical Systems (CPS), changes the challenges to the application of MLOps. The aim of this paper is to discuss challenges of applying MLOps in an Industry 4.0 context. As a result we identify challenges to MLOps that are specific to the Industry 4.0 context or to specific scenarios within Industry 4.0 (specific challenges) and challenges that are roughly comparable to MLOps in other contexts (general challenges). This serves as a basis for finding solutions to address the identified novel challenges. Hence, this paper aims at researchers working on platforms in an Industry 4.0 environment.

The paper is structured as follows: "Related Work" provides an overview of the related work. "MLOps in Industry 4.0" introduces our understanding of MLOps, which relies on existing models, but with an adaptation to the Industry 4.0 context. "Challenges" is the core of the paper and presents the challenges we could identify. We discuss these in an integrated manner in "Discussion", while "Conclusion" concludes the paper.

This article is part of the topical collection "Innovative Intelligent Industrial Production and Logistics 2022" guest edited by Alexander Smirnov, Kurosh Madani, Hervé Panetto and Georg Weichhart.

✉ Leonhard Faubel
faubel@sse.uni-hildesheim.de

✉ Klaus Schmid
schmid@sse.uni-hildesheim.de

Holger Eichelberger
eichelberger@sse.uni-hildesheim.de

[1] Software Systems Engineering, Institute of Computer Science, University of Hildesheim, Universitätspl. 1, 31141 Hildesheim, Lower Saxony, Germany

Published online: 28 October 2023

Related Work

We conducted a systematic literature review following the guideline proposed by Kitchenham et al. [4] to find the typical activities related to MLOps. The method used is described in detail in a technical report (review protocol) [5]. Included are peer-reviewed and published research studies, such as conference and journal papers using the term MLOps or Machine Learning Operations. We also included articles where the terms above were not the main or only purpose of the article. All entries from January 2015 to May 2022 were considered here. Excluded are duplicate versions of studies, studies in other languages than English, not peer-reviewed studies, books, grey literature, studies that just mention MLOps or Machine Learning Operations without explanation or using it, and studies without reference to the MLOps activities, as well as talks without available information like protocols or notes, and posters. The corresponding search engines are listed in Table 1 together with dates of the searches. ACM Digital Library is a full-text collection of articles published by the Association of Computing Machinery (ACM), including all magazines and conference articles. Further, it contains a bibliographic database containing publications from all major publishers of computing literature [6]. IEEE Xplore digital library allows for discovery and access to journal and conference papers on computer science, electrical engineering, and electronics. It is a research database for articles published by the Institute of Electrical and Electronic Engineers (IEEE) and other publishers [7]. Science Direct allows searching for scientific and medical publications. It provides access to a bibliographic database of the publisher Elsevier [8]. These digital libraries were selected because they are the biggest and most common search engines for publications in software engineering.

The general search process began with the definition of the review protocol [5]. Then, the search engines were selected. Keywords were defined that could be used to find the subject of the topic. The papers found were filtered based on inclusion and exclusion criteria and then used to identify MLOps activities.

After an initial search, we used specific keywords, which are combined to form a search query based on the scope of the literature review: *("MLOps" OR "ML Ops" OR "ML-Ops" OR "Machine Learning Operations").*

MLOps are defined in Sect. "MLOps Definition". Its activities are described in Sect. "MLOps Activities", lifecycles in Sect. "MLOps Life-Cycles", and automation in Sect. "Automation".

Table 1 Search engines used for the systematic literature review and the corresponding search dates

Search engines	Search date
ACM Digital Library	2022/05/31
IEEE XPlore	2022/05/24
Elsevier/Science Direct	2022/05/31

MLOps Definition

MLOps is specialized DevOps principles for the ML application domain [2, 9–11]. DevOps describes practices that integrate software development with IT system operations. It uses experiences from these two areas for continuous improvement of the quality of software systems. Further, it reduces costs and time to deployment. Especially, DevOps provide insight from operations during development [9].

MLOps is frequently described as a collection of techniques and tools, practices, or processes for ML deployment in production [2, 12, 13]. In addition, MLOps is designed to significantly lower the time-to-delivery [11, 14, 15]. The deployment process can be done manually or automatically [14]. However, to reduce the time-to-delivery, MLOps aims to increase the level of automation [10, 16, 17]. A very important property often associated with MLOps is the reproducibility and repeatability of the models [10, 18]. This creates the need for versioning of data, models, code and configurations [19–21]. CI (continuous integration), CD (continuous deployment) [10, 22], and CT (continuous training) [23–25] are considered to be fixed components of MLOps [11, 12, 26]. In CI, changes made by developers are continuously integrated into a repository. A project build is then automatically executed, and the changes are integrated into the code via CD and published to the production environment [9]. CT provides automated re-training of ML models [23].

MLOps Activities

The activities most frequently used related to MLOps are data collection, data analysis, data preparation, model building, model training, model evaluation, model selection, model packaging, model deployment, and model monitoring. Their appearance in the literature is shown in Table 2. In addition, the following terms appear sporadically: planning/analysis and design [23], requirements engineering [23], data cleaning [19], feature engineering [27–29], divide the data into training, testing, and cross-validation sets [11, 25], hyperparameter tuning [11, 25]/optimization [10, 29], model registering [30], algorithm configuration [10, 27], (code) testing [10, 11, 25], (system) integration [24, 25], releasing [25], infrastructure management [25], output production [28, 31]/operate [10]/inference [16], versioning [10, 19–21].

Table 2 MLOps activities

Study	Year	Data collection	Data analysis	Data preparation	Model building	Model training	Model evaluation	Model selection	Model packaging	Model deployment	Monitoring
[2]	2019	X	X	X	X	X				X	X
[27]	2020	X	X	X			X	X			X
[13]	2020				X	X	X			X	
[31]	2020	X	X	X		X				X	
[19]	2020	X		X		X				X	
[10]	2021	X	X	X	X	X	X		X	X	X
[28]	2021	X	X	X		X				X	
[23]	2021	X			X		X			X	X
[11]	2021	X				X	X	X		X	X
[14]	2021				X	X			X	X	X
[30]	2021	X	X			X	X		X	X	X
[32]	2021	X								X	
[25]	2021	X				X	X	X		X	X
[29]	2022	X	X	X			X	X	X	X	X
[16]	2022	X	X	X		X	X			X	X
[33]	2022	X		X		X	X			X	
[17]	2022	X	X	X		X				X	X
[21]	2022	X		X		X			X	X	X
[12]	2022	X	X	X	X		X	X	X	X	X

MLOps Life-Cycles

Various MLOps cycles can be found in the literature. These have similarities, as described above, but also differences to the traditional DevOps life-cycle. However, the ML cycle can be combined with the DevOps cycle with some modifications to provide a CI/CD workflow for intelligent applications [13]. The two following life-cycles of MLOps are described in the literature, highlighting the relation to DevOps. The MLOps life-cycle used by Symeonidis et al. includes three phases: ML, development, and operations [12]. Van der Goes uses a variant with four stages [10]. Here, the ML stage is divided into data management and modeling. Each stage consists of a cycle with tasks that connect the cycles. A very similar life cycle with four stages is presented by Tamburri et al. [31]. It also includes data as a separate cycle, but differs in the activities included. Another MLOps life-cycle by Ranawana et al. [23] describes a life-cycle which forms a pipeline with manual and automated steps and a feedback loop from the last step "Operate, Monitor & Maintain" to the step "Analysis & Design".

Automation

ML encompasses different technologies, algorithms, tools, and libraries. For this reason, a product pipeline is made up of a series of connections ranging from hardware to software, from raw data storage to the dissemination of information, from web services to endpoint software. It is unrealistic to manage all of these pieces manually [33]. To simplify the handling of this complexity, MLOps aims at automation. Here, DevOps principles and approaches are applied to automate ML activities [12]. Well-known cloud providers such as *Microsoft Azure*, *Google Cloud* and *Amazon AWS* provide ready-made solutions. However, automation also happens in-house. Tools such as *DVC*, *Airflow*, and *Git* are used here.

MLOps helps develop and deploy all ML development steps and supports deployment [25]. It often automates model training, evaluation, deployment and monitoring [23]. Further, it includes integration, test, release, deployment and infrastructure management [11]. Consequently, there are different types of automation. On the one hand, it is possible to automate the ML pipeline end-to-end [11]. In addition, the ML steps can be automated, for example, to independently search for suitable ML methods, hyperparameters and configurations [16, 28]. Furthermore, it is possible to automate the infrastructure management [28].

Pipeline automation: An end-to-end pipeline is a pipeline for process automation [21]. By pipeline automation, we mean automatic actions to avoid manual activities and delays. For this purpose, the preceding ML steps must be started automatically, which are required for the subsequent model deployment [13, 14]. CI, CD, and CT are used to

make ML algorithms run automatically [16, 28, 34]. For example, Bash and Python scripts or process automation software can be used to implement process automation [20]. Today, many tools help automate the ML pipeline [12].

MLOps-step automation: If automatic retraining is required as part of CT, an automatic trigger is needed to start the ML pipeline. For example, data and models may change, resulting in a worse outcome. Automatic data and model monitoring can detect this at an early stage. For example, a threshold in model performance can trigger retraining [2, 30]. If such a correlation does not exist, e.g., customer feedback can be used as a trigger [35]. Appropriate functions must be provided for this purpose. AutoML can further automate individual ML steps. For example, an AutoML pipeline can implement the steps of data preparation, model creation, hyperparameter tuning, evaluation, and validation [12]. Iterative learning algorithms automatically tune the control parameters in this process [36]. Newly available technologies also increasingly enable automatic ML feature selection, data labeling, and model generation [23].

Infrastructure management: Some tools handle iterative tasks that take a long time and allow the creation of models with high scalability, efficiency, and productivity, while maintaining quality [20]. Thus, they can perform tasks of the pipeline components. These components of the pipeline can be deployed on devices on-site or in the cloud [28, 30]. The deployment often depends on the performance needs and the hardware used in the specific use case. For this reason, tools and platforms are required that can handle the necessary hardware needs. These tools often depend on cloud platforms and are offered as Software as a Service (SaaS) models. Further, infrastructure-as-code can help manage and provision the infrastructure with physical and virtual servers [20]. Container technology that packs code, libraries, and configuration files for a deployment helps to realize this kind of cloud deployment. The containers can be orchestrated and coordinated using scripts or special tools [9].

MLOps in Industry 4.0

MLOps aims to enhance the automation and quality of intelligent systems [26]. It combines principles from DevOps with machine learning. The flexibility provided by DevOps principles is beneficial to machine learning (ML) as, typically, several iterations need to occur to identify well-working ML models and then adapt them over time as the situation changes in the application.

High-Level Structure of the MLOps Cycle

Figure 1 represents a high-level structure of the MLOps cycle. This structure is divided into several steps: a manual step (Sect. "Manual Step"), an MLOps/Automation step (Sect. "Automation/MLOps"), and a model application step (Sect. "Model Application"). The steps are explained in detail below.

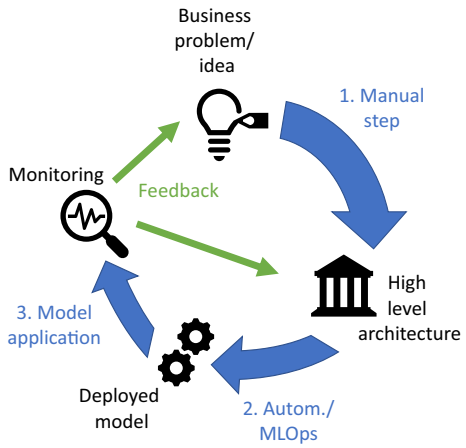

Fig. 1 High-level structure of the MLOps cycle [37]

Manual Step

The MLOps cycle typically begins with a business problem to be solved by ML. The initial step in MLOps to solve this problem is always manual and is performed in an analysis environment. This allows to first understand the problem and solution possibilities using ML. We investigate whether and how the problem can be solved and which algorithms work best to solve the problem. Usually, initial data analysis requires significant amounts of data and powerful hardware for initial experiments with different ML methods. In this initial step some ML tasks can be automated with autoML and hyperparameter tuning on the experimental hardware.

Based on the results of this step, a high-level architecture for the MLOps solution is created. This includes considerations of the infrastructure in the production environment and related requirements. We must adapt the ML model to these requirements. This takes the ML method, SE architecture, hardware architecture, configuration, and, if applicable, even the architecture of the CPS into account and may even evolve them, if necessary. In particular, this defines the deployment of the various MLOps components and under which conditions they are triggered.

Automation/MLOps

Then, we automate the pipeline to solve our problem and deploy it to the production environment. One input to the high-level architecture definition is whether automated retraining of models should be supported as well (Step 2). On the one hand, this depends on the outcome of the model

development; on the other, it depends on business decisions. For example, in critical business cases automatic retraining could be a problem. If ML is used for control and automatic retraining is conducted without human intervention, the behavior of a system may change in unexpected and negative ways. Also, the type of algorithms identified in the model development significantly influence the high-level system architecture, e.g., neural networks have very different resource implications vs. random forest classification.

A key step in MLOps is the ramping up of ML into production. Before that, it is crucial to versioning data, models, code, and configuration to make the ML models repeatable and traceable. Continuous integration and deployment bring ML models smoothly into production. Integration can also include testing such as A/B and shadow testing.

Model Application

Finally, there is always a model application stage (Step 3), which is often performed on edge devices. The details of the architecture will have a strong influence on the hardware resources as well as the ML components used, as we will discuss in Sect. Challenges.

Usually, the ML model is monitored in production. The purpose of monitoring is to determine whether the model is still working properly. This involves ML performance measurements such as model precision and accuracy as well as monitoring for changes in the input data like data drift. Based on monitoring information, automated model adaptations can be triggered (automated MLOps) or it can be signaled that such adaptations may need to be done manually, leading to a redefinition of the machine learning approach. For this purpose, a threshold can be used to determine if re-training is necessary.

In some cases, when re-training has no (positive) effect, it is necessary to go back to the business problem and rethink the model from scratch including the high level architecture.

MLOps Model

While Fig. 1 provides a high-level overview of MLOps in Industry 4.0, we need a more detailed MLOps model to define the individual challenges. Various life-cycle models have been proposed for MLOps [10]. Two MLOps life-cycles are predominant in the literature. Symeonidis et al. depict an MLOps life-cycle with three stages: ML, development, and operations [12]. Van der Goes describes a variant with four stages [10]. Here, the ML stage is subdivided into data management and modeling. Each stage consists of a cycle with tasks. These tasks connect the cycles to each other.

However, these models do not address the specific activities of MLOps, but make clear that ML is added to DevOps principles. Moreover, they do not include cross-cutting activities. Therefore, we propose a new life-cycle.

Our lifecycle model is based on the activities identified in the related works in Table 2. It defines the phases (Fig. 2): Data Engineering, Model Engineering, and Operations, each consisting of several activities; they are complemented by supporting activities. These phases, together with associated activities are described below. They all depend on the overarching high level architecture in the industrial environment.

Data Engineering

Data engineering comprises the initial activities that are exclusively related to data. These include data collection, data analysis, and data preparation. Still, these activities are often among the most time-consuming.

Data collection: This provides a basis for machine learning. The collected data can include machine data, product information, customer data, behavioral data, etc. Data can be collected, e.g., by web scraping, counting actions in a process, or querying records relevant to the business case, such as required information on the factory floor. The relevant data needs to be determined based on the use case and typically provides insights into the effectiveness and quality of the production process.

Data analysis: This step aims at understanding the data and its quality, e.g., by identifying outliers. The data analysis aims to extract meaningful insights and patterns. This is a crucial step in developing and deploying machine learning models, as it helps to identify critical features and variables that can be used to train the models effectively. Typically, statistical and ML methods are used. Ultimately, data analysis is essential for ensuring machine learning models' accuracy, reliability, and effectiveness.

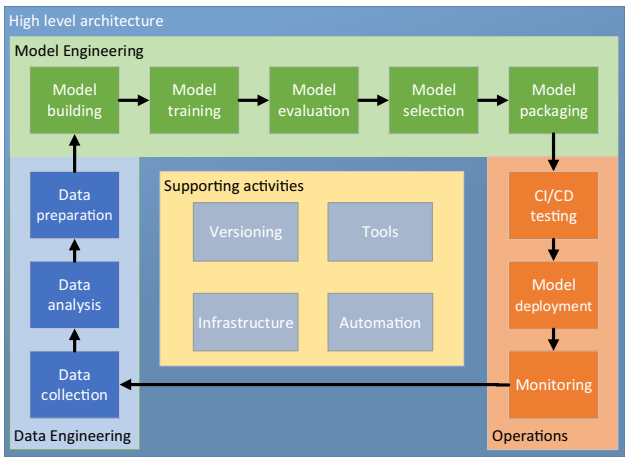

Fig. 2 MLOps activities [37]

Data preparation: Transformations are applied, e.g, for data cleaning or value imputation. Feature transformation can also be done here [27].

Model Engineering

Model Engineering includes all steps that primarily serve to create a model. It ends when the model is tested and deployed. In the structure given in Fig. 1 this can either be performed manually (especially in the first iteration) or in an automated way. Typically, an ML pipeline is created at this stage so that when repetitive operations are performed manually, there are no potential errors that can slow down the MLOps process. Model Engineering comprises model building, model training, model evaluation, model selection, and model packaging.

Model building: Model building aims at creating the necessary machine learning models. This includes the identification of the relevant approaches (e.g., neural networks vs. decision trees), determining corresponding model structures, and potentially determining hyperparameters.

Model training: Candidate ML models are trained and fitted to the data.

Model evaluation: ML models are evaluated on test data [33].

Model selection: The most appropriate (usually the best best performing) ML model is selected (or multiple models, if there are several problems, which are addressed by ML techniques). Potentially further fine-tuning is performed [27].

Model packaging: The final ML models are packaged as one or more application components or as a "model as a service" [38].

Operations

MLOps involves thinking about the quality of the model in the real world upfront. The operation phase, which is described here, must take that into account. It includes CI/CD testing, model deployment, and monitoring activities.

CI/CD-Testing: The ML model can be integrated into the code and tested before deployment to get insights in advance whether and how it works in production. As part of continuous integration and deployment, special tests for features, data, models, ML infrastructure, and monitoring for ML are run to ensure quality of the deployed system.

Model deployment: The production-ready ML models are deployed as part of the production system [29].

Monitoring: Real data may behave differently than a training set. Since data in the real world can change constantly, models may stop working after some time. The performance (quality) of the ML models is continuously monitored to determine whether a manual or automated intervention is necessary [29]. Although ML methods are highly appreciated in Industry 4.0 settings, according to our experience, the application in production settings is sometimes a bit conservative, i.e., automatically adjusting an existing or deploying a new version of a working ML model is judged rather sceptically. Thus, at least the option to manually approve such an intervention is often requested.

Supporting Activities

There are also supporting activities. These are a prerequisite for one or more of the activities, or for an MLOps implementation. These activities include providing infrastructure components, versioning, automation, and providing tools.

Infrastructure: The necessary infrastructure components, including the relevant hardware, required to develop, deploy, and run complex ML systems must be selected, acquired, installed, and maintained. Often, this also involves data integration activities, e.g., in Industry 4.0 settings to obtain additional data from ERP (Enterprise Resource Planning) or MES (Manufacturing Execution System) systems.

Versioning: When experimenting, it may be unclear which models were trained with which data, features, and configurations to obtain a particular result. For this reason, versioning of the code, data, ML model, and configuration details is required [20]. In addition, versioning is often desirable to revert to the last working model to avoid or at least minimize (production) downtime. Versioning is also considered an ML safety mechanism, especially when manual approvals of modified ML models are required. In this case, an updated version of the ML model can be tested, and the old version can be easily restored if not approved.

Automation: Various steps in the overall lifecycle are often automated. This requires the implementation of these steps, either using existing automation capabilities or implementing them in special ways.

Tools: Tools for developing ML applications are needed. This includes domain-specific tools like domain-oriented modeling or simulation (e.g., for factories or machines in an Industry 4.0 scenario).

Challenges

This section presents challenges relevant to using MLOps in an Industry 4.0 environment. We identified them based on our project experience as well as based on literature related to the described tasks. The challenges are organized based on the MLOps activities model in Fig. 2. We distinguish between data engineering, model-engineering, operations, monitoring, and support-activity-related challenges. Challenges exist in all of these activity groups. The focus of our discussion is always: which activity does not

cause additional difficulties, and what additional difficulties exist in MLOps for Industry 4.0 over more traditional MLOps scenarios. Of course, this may vary depending on project-specific requirements. The identified challenges are divided into general and specific challenges. General challenges also exist in situations in other contexts, while specific challenges relate exclusively to specific situations in Industry 4.0 scenarios.

Data-Related Tasks

Here, we describe the challenges in the data-related MLOps tasks.

Data collection: Depending on the scenario, various technical challenges exist. Data acquisition requires suitable sensors and data transmission in the factory environment as data collection starts in the manufacturing machine and ends in the software system. Depending on the application, real-time requirements exist and large amounts of data are transmitted or stored. We need specific information in accurate time intervals delivered by sensors and transmitted to the ML component.

If suitable technical conditions exist, i.e., machines with the appropriate equipment, in which the sensors and the respective networking are adequately dimensioned, applying MLOps is more straightforward, as one only has to think about potential problems in storing the relevant amounts of data.

Otherwise, problems may arise in meeting the requirements for the analysis. In this case, data collection for MLOps becomes very difficult as converting or adapting existing hardware and software may involve a significant effort. For example, adding additional sensors to a machine is a non-trivial task. In a worst-case scenario, whole machines, infrastructures, or manufacturing lines must be redeployed or exchanged.

Data analysis: Data analysis involves the extraction of meaningful insights and patterns, which are dependent on domain knowledge. Domain knowledge is usually required to understand underlying processes and correlations in the data. Typically, this is performed with a sample data set outside the Industry 4.0 environment. Thus, there is no unusual complexity in this task.

Data preparation: Features are single independent variables in data that serve as input to an ML system. These features usually describe characteristics of the production process. In data preparation, features are provided online for inference and offline for experiments and training. This is often implemented in the form of feature stores to realize reproducibility and versioning using feature and metadata registries [3]. Before feature selection, it is common to deal with data in the Big Data range. After the selection, which takes place during pre-processing, the data sets to be processed are usually smaller. We need hardware on the factory floor that is fast enough to pre-process the data.

The following specific challenges arise: Real-time requirements typically apply when data preparation is done as a pre-processing step in model inference in production. In this case, the frequency and size of samples required for further steps and the capabilities of existing hardware influence how challenging this activity is. High-quality hardware and software measures may be required in the production environment with high real-time requirements. However, the applicability and availability of such hardware may be limited by technical factory floor regulations, e.g., electrical or mechanical norms, as well as by budget restrictions. Further, factory floor regulations include rules like data protection laws, internal company regulations, or even relays for companies with an increased demand for protection. Sometimes customers do not want a change in their existing hardware infrastructure because a production line must not be interrupted, and possibly the maintenance and repair complexity should not increase.

Model-related Tasks

Here, we describe the identified challenges in the model-related MLOps activities.

In general, if the model building, model training, model evaluation and model selection activities are part of an automated pipeline, there is a need for additional software, hardware, and infrastructure considerations (see Sect. "Support Activities" below).

Model building, training, evaluation, and selection: Automated training is increasingly required. If this takes place outside the Industry 4.0 environment, which is often the case, there is no impact on regular operation. However, due to the need of retraining, there is also in performing these steps within a factory environment. In this case additional hardware resources and software infrastructures like GPUs or ML implementations for embedded devices are needed within the factory.

Model packaging: In model packaging, the particular constraints of the available infrastructure in the factory like hardware resources must be taken into account. The hardware resources can be special hardware like GPU/TPU/FPGA. Also, the network bandwidth, operating systems and storage capacities are sometimes limited, adapted or unique for the use in the production environment.

Operations-related tasks: The following challenges address operations-related tasks that relate to activities required for deployment and during ongoing operations in the MLOps environment.

CI/CD testing: As large parts of a CI/CD testing environment will be in a classical IT-environment, no special challenges stem from this part. However, there is a general

challenge: the heterogeneity of hardware and operating systems typical of Industry 4.0 can make these tasks more complex than usual, which leads to additional tests.

Also, intelligent methods require special CI/CD tests. These include tests for features and data, model development, ML infrastructure, and monitoring tests for ML serving, e.g., performance. A particular challenge is that ML elements are typically analyzed on a statistical basis, while traditional testing requires correctness of each individual test case. Moreover, in automated adaptation scenarios, even these tasks may happen within a factory environment, making this a rather complex task.

Model deployment: MLOps activities can be performed on the factory floor, in the cloud, or in corporate IT environments. Specifically, in the case of the factory floor, edge resources are needed, leading to the typical problems of sufficient and appropriate resources to ensure necessary technical performance requirements.

Monitoring: In Industry 4.0, monitoring physical processes is an integral part. Ideally, the existing monitoring solution is suitable for MLOps or can be extended easily. If this is not the case, e.g., if physical separation of groups or networks is required, a specific challenge arises. Additional hardware requirements or development efforts become necessary.

Support Activities

Support Activities are cross-cutting activities related to infrastructure, tools, versioning and automation. Here, we address challenges referring to support activities.

Infrastructure: A major, general challenge related to the infrastructure is heterogeneity, which is even more significant than for MLOps in information systems. Some parts need to run in an embedded context, some in corporate IT environments. Embedded devices are devices that contain a particular purpose computing system. Corporate IT may have restrictions on IT that cannot be bypassed. In case that automates model engineering activities to some extent, there even exists the case that for the same steps, multiple different infrastructures are needed (IT vs. embedded).

Hybrid cloud infrastructures offer numerous benefits. Companies can keep confidential data on-premise while taking advantage of the cloud, e.g., the ability to scale resources as needed. Further, with a hybrid cloud setup, businesses can quickly and easily spin up additional resources when demand increases without investing in hardware or software. This reduces the risk of over-provisioning.

Existing computing frameworks like TensorFlow are typically unavailable as implementations on the factory floor. Rather, this requires special frameworks available for edge computing that may not bring the same features, e.g., TensorFlow-light. The higher the level of automation and the requirements for the properties associated with MLOps become, the more difficult it is to deploy them in the environment of IoT and edge devices. This leads to adapted frameworks for specific use in particular situations.

Another general problem is the lack of widely and homogeneously adopted standards, which is partially due to (expensive) legacy machines or retrofitting of factory equipment. Legacy machines do not support communication with other systems. This also leads to a lack of standardization of tools, making tool selection in the context of Industry 4.0 a significant challenge.

The extent to which parts of the MLOps cycle are automated varies significantly among cases. Of course, several steps, like deployment or productive operation, are usually automated.

We envision that some degree of automated adaptation will also become standard practice in Industry 4.0. Nevertheless, difficult questions will remain about what may be changed independently in the model during productive operation. In particular, changes by self-adaptation may impact hardware requirements and reliability.

MLOps emphasizes the management of interdependencies among data, models, and code. Thus, versioning this information is essential. If this information should be available at edge nodes, e.g., as feature stores, appropriate versioning infrastructure must also be available there.

Tools: Tools are a general challenge. Typically, a more complex tool environment is required for MLOps in Industry 4.0. This is due to the fact that some steps will be done manually in an IT-environment, but also corresponding tools in the embedded environment are needed. Also, tools that address the specifics of the industry environment (e.g., cross-compilation) are required. Special tools like simulation environments may also be needed to study the impact of ML solutions. In particular, if they influence the factory behavior.

Discussion

Typically, existing MLOps life-cycle models do not describe that some activities can be either performed manually or in an automated way at different points in time. With our revised model of MLOps, we tried to capture this.

An essential part of MLOps in Industry 4.0, which is typically not present in other MLOps models, is the definition of a high-level architecture. This is particularly relevant as a connection to software engineering. Hence, we introduced this here.

Applying MLOps principles in an Industry 4.0 context is not very specific. Challenges exist mainly for the reason that it is a heterogeneous environment, and many individual solutions are involved. A full-scale architecture

and implementation must cover the whole environment of the cyber-physical system, or at least interface with relevant existing solutions in this context. This is particularly challenging due to severe resource constraints on the embedded devices and the complexity introduced due to the many different hardware platforms and operating environments and increasingly distributed computing. If self-adaptation is introduced and corresponding model engineering happens on edge devices, the complexity of the environment becomes even more severe. Additionally, it remains unclear when and under what conditions a high-level architecture needs to be re-built or when a re-learning of the existing pipeline is sufficient.

The majority of MLOps activities are of similar complexity in an Industry 4.0 case as in information systems. This counts for activities used with a CPS, as well. The main reason is that they are typically performed outside the factory infrastructure, especially if not automated. This applies to "Data analysis" and the majority of model-related tasks.

The starting level of the technical infrastructure and the aimed degree of automation influence the overall complexity of implementing MLOps in Industry 4.0. The Infrastructure is complex and involves heterogeneity and individual solutions. Challenges arise, especially in not-model-related activities under certain conditions. It is highly case-dependent whether MLOps is challenging to implement in Industry 4.0. For example, technical challenges arise in data-related tasks when already existing devices are not ready for data collection for the specific use case of an ML model or its execution. Operations are problematic when particular implementations are required, or more resources are needed.

Some automated adaptation scenarios may become standard practice and ease ML activities. For CI/CD Testing, the existence of digital twins may be leveraged using existing virtualization and simulation capabilities to test the models.

If applied, standardization activities like OPC UA companion specs [39], and Asset Administration Shells [40] may ease data collection and analysis in the future by providing well-defined interfaces in addition to precise semantics. Nevertheless, these technologies are not widely adopted.

Today, some guidance, best practices, frameworks, and platforms already exist to facilitate MLOps in IoT environments (and thus Industry 4.0): Ruf et al. discuss a selection of benefits using MLOps in industrial scenarios [41]. A digital twin architecture with MLOps techniques is proposed by Fujii et al. [42]. An MLOps framework for automated ML at the edge is described by Raj et al. [30]. None of these takes a broad view of problems in MLOps in Industry 4.0 as we do here.

Conclusion

MLOps is an important set of practices and activities, which are key to the implementation of modern ML-based software solutions. It is also strongly related to DevOps principles, making it all the more important at the intersection with software engineering. MLOps in Industry 4.0 has, however, not yet received the necessary level of attention.

In this paper, we discussed challenges from the perspective of MLOps in Industry 4.0 and how they differ from MLOps challenges in other contexts. Overall, we conclude that most Industry 4.0 MLOps challenges exist in a similar manner in the more traditional software engineering context. Some additional challenges exist, at least in some application scenarios. In particular, we could identify significant (specific) challenges for four activities. Most of the challenges are not unique to Industry 4.0, a positive indicator for using existing technologies and practices in this context as well. We plan to study these and the corresponding ways to address them in more detail in the future. For this purpose, we will conduct industry studies around MLOps and apply what we have learned to platforms and frameworks that implement MLOps for Industry 4.0.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was partially supported by EXPLAIN (01IS22030E) funded by the Federal Ministry of Education (BMBF) and IIP-Ecosphere funded by the German Ministry of Economics and Climate Action (BMWK) under grant number 01MK20006C.

Data availability The raw data supporting this study's findings were generated at the University of Hildesheim. Derived data supporting the findings of this study are available from the corresponding author, Leonhard Faubel, on request.

Declarations

Compliance and Ethical Standards All applicable international, national, and/or institutional guidelines for Software Engineering were followed. No human participants, human material, human data, or animals were involved in this research.

Conflict of Interest All authors declare that they have no conflicts of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Borgmeier A, Grohmann A, Gross SF. Smart Services und Internet der Dinge: Geschäftsmodelle, Umsetzung und Best Practices: Industrie 4.0, Internet of Things (IoT), Machine-to-Machine, Big Data, Augmented Reality Technologie. Carl Hanser, München, Germany 2017.
2. Poss C, Irrenhauser T, Prueglmeier M, Goehring D, Zoghlami F, Salehi V, Ibragimov O. Enabling robot selective trained deep neural networks for object detection through intelligent infrastructure. In: Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering. Association for Computing Machinery, New York, USA 2019. https://doi.org/10.1145/3351917.3351982.
3. Kreuzberger D, Kühl N, Hirschl S. Machine learning operations (MLOps): overview, definition, and architecture. Preprint at https://doi.org/10.48550/arXiv.2205.02302 2022.
4. Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S. Systematic literature reviews in software engineering—a systematic literature review. Inform Softw Technol. 2009;51(1):7–15. https://doi.org/10.1016/j.infsof.2008.09.009. (**ISBN: 0950-5849 Publisher: Elsevier**).
5. Faubel L, Schmid K. Review protocol: a systematic literature review of MLOps. Hildesheimer Informatik Berichte (1/2023, SSE 2/23/E) 2023.
6. About ACM DL. 2023. https://dl.acm.org/about accessed 2 Jan 2023.
7. About Content in IEEE Xplore. 2021. https://ieeexplore.ieee.org/Xplorehelp/overview-of-ieee-xplore/about-content. Accessed 2 Jan 2023.
8. ScienceDirect.com Science, health and medical journals, full text articles and books. 2023. https://www.sciencedirect.com. Accessed 2 Jan 2023.
9. Capizzi A, Distefano S, Mazzara M. From devops to devdataops: data management in devops processes. In: Bruel J-M, Mazzara M, Meyer B, editors. Software engineering aspects of continuous development and new paradigms of software production and deployment. Cham: Springer; 2020. p. 52–62. https://doi.org/10.1007/978-3-030-39306-9_4.
10. Goes M. Scaling enterprise recommender systems for decentralization. In: Proceedings of the 15th ACM Conference on Recommender Systems. Association for Computing Machinery, New York, USA 2021; pp. 592–594. https://doi.org/10.1145/3460231.3474616.
11. Mäkinen S, Skogström H, Laaksonen E, Mikkonen T. Who needs mlops: what data scientists seek to accomplish and how can mlops help? In: 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), 2021: pp 109–112. https://doi.org/10.1109/WAIN52551.2021.00024.
12. Symeonidis G, Nerantzis E, Kazakis A, Papakostas GA. MLOps—definitions, tools and challenges. In: 12th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, Las Vegas, NV, USA 2022, pp. 0453–0460. https://doi.org/10.1109/CCWC54503.2022.9720902.
13. Liu Y, Ling Z, Huo B, Wang B, Chen T, Mouine E. Building a platform for machine learning operations from open source frameworks. IFAC-PapersOnLine. 2020;53:704–9. https://doi.org/10.1016/j.ifacol.2021.04.161.
14. Garg S, Pundir P, Rathee G, Gupta PK, Garg S, Ahlawat S. On continuous integration/continuous delivery for automated deployment of machine learning models using MLOps. In: Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). IEEE, Laguna Hills, USA, 2021; pp 25–28. https://doi.org/10.1109/AIKE52691.2021.00010.
15. Gupt KK, Raja MA, Murphy A, Youssef A, Ryan C. GELAB—the cutting edge of grammatical evolution. IEEE Access. 2022;10:38694–708. https://doi.org/10.1109/ACCESS.2022.3166115.
16. Brik B, Boutiba K, Ksentini A. Deep learning for b5g open radio access network: evolution, survey, case studies, and challenges. Open J Commun Soc. 2022;3:228–50. https://doi.org/10.1109/OJCOMS.2022.3146618.
17. Mosqueira-Rey E, Pereira EH, Alonso-Ríos D, Bobes-Bascarán J. A classification and review of tools for developing and interacting with machine learning systems. In: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing. SAC '22. Association for Computing Machinery, New York, USA, 2022; pp 1092–1101. https://doi.org/10.1145/3477314.3507310.
18. Ismail BI, Khalid MF, Kandan R, Hoe OH. On-premise AI platform: from DC to edge. In: Proceedings of the 2019 2nd International Conference on Robot Systems and Applications. ICRSA 2019. Association for Computing Machinery, New York, USA, 2019; pp 40–45. https://doi.org/10.1145/3378891.3378899.
19. Zhou Y, Yu Y, Ding B. Towards mlops: a case study of ml pipeline platform. In: 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), 2020; pp 494–500 . https://doi.org/10.1109/ICAICE51518.2020.00102.
20. Oluyisola OE, Bhalla S, Sgarbossa F, Strandhagen JO. Designing and developing smart production planning and control systems in the industry 4.0 era: a methodology and case study. J Intell Manuf. 2022;33:311–32. https://doi.org/10.1007/s10845-021-01808-w.
21. De Silva D, Alahakoon D. An artificial intelligence life cycle: from conception to production. Patterns. 2022. https://doi.org/10.1016/j.patter.2022.100489.
22. George J, Saha A. End-to-end machine learning using kubeflow. In: 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD). CODS-COMAD 2022. Association for Computing Machinery, New York, NY, USA, 2022; pp 336–338. https://doi.org/10.1145/3493700.3493768.
23. Ranawana R, Karunananda AS. An agile software development life cycle model for machine learning application development. In: 5th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI). IEEE, Melbourne, Australia, 2021; pp. 1–6 https://doi.org/10.1109/SLAAI-ICAI54477.2021.9664736.
24. Karácsony T, Loesch-Biffar AM, Vollmar C, Noachtar S, Cunha JPS. DeepEpil: towards an epileptologist-friendly AI enabled seizure classification cloud system based on deep learning analysis of 3d videos. In: IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), 2021; pp 1–5. https://doi.org/10.1109/BHI50953.2021.9508555.
25. Granlund T, Kopponen A, Stirbu V, Myllyaho L, Mikkonen, T. MLOps challenges in multi-organization setup: experiences from two real-world cases. In: 1st Workshop on AI Engineering—Software Engineering for AI (WAIN). IEEE/ACM, Madrid, Spain, 2021; pp 82–88. https://doi.org/10.1109/WAIN52551.2021.00019.
26. Meedeniya D, Thennakoon H. Impact factors and best practices to improve effort estimation strategies and practices in devops. ICICM '21. Association for Computing Machinery, New York, USA, 2021; pp 11–17. https://doi.org/10.1145/3484399.3484401.
27. Cardoso Silva L, Rezende Zagatti F, Silva Sette B, Santos Silva L, Lucrédio D, Furtado Silva D, Medeiros Caseli H. Benchmarking machine learning solutions in production. In: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020; pp 626–633. https://doi.org/10.1109/ICMLA51294.2020.00104.
28. Li Z, Liu X-Y, Zheng J, Wang Z, Walid A, Guo J. Finrl-podracer: high performance and scalable deep reinforcement learning for quantitative finance. In: Proceedings of the Second ACM

International Conference on AI in Finance. ICAIF '21. Association for Computing Machinery, New York, USA (2022). https://doi.org/10.1145/3490354.3494413.
29. Rahman S, Kandogan E. Characterizing practices, limitations, and opportunities related to text information extraction workflows: A human-in-the-loop perspective. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. CHI '22. Association for Computing Machinery, New York, USA (2022). https://doi.org/10.1145/3491102.3502068.
30. Raj E, Buffoni D, Westerlund M, Ahola K. Edge MLOps: an automation framework for AIoT applications. In: International Conference on Cloud Engineering (IC2E). IEEE, San Francisco, USA, 2021; pp 191–200. https://doi.org/10.1109/IC2E52221.2021.00034.
31. Tamburri DA. Sustainable mlops: trends and challenges. In: 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2020; pp 17–23. https://doi.org/10.1109/SYNASC51798.2020.00015.
32. Borg M, Jabangwe R, Åberg S, Ekblom A, Hedlund L, Lidfeldt, A. Test automation with grad-CAM heatmaps—a future pipe segment in MLOps for vision AI? In: International Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, Porto de Galinhas, Brazil, 2021; pp 175–181. https://doi.org/10.1109/ICSTW52544.2021.00039.
33. ...Sun Z, Sandoval L, Crystal-Ornelas R, Mousavi SM, Wang J, Lin C, Cristea N, Tong D, Carande WH, Ma X, Rao Y, Bednar JA, Tan A, Wang J, Purushotham S, Gill TE, Chastang J, Howard D, Holt B, Gangodagamage C, Zhao P, Rivas P, Chester Z, Orduz J, John A. A review of earth artificial intelligence. Comput Geosci. 2022. https://doi.org/10.1016/j.cageo.2022.105034.
34. Akinosho TD, Oyedele LO, Bilal M, Barrera-Animas AY, Gbadamosi A-Q, Olawale OA. A scalable deep learning system for monitoring and forecasting pollutant concentration levels on UK highways. Ecol Inform. 2022. https://doi.org/10.1016/j.ecoinf.2022.101609.
35. Yasser A, Abu-Elkhier, M: Towards fluid software architectures: bidirectional human-AI interaction. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2021; pp 1368–1372. https://doi.org/10.1109/ASE51524.2021.9678647.
36. Fernando H, Marshall J. What lies beneath: material classification for autonomous excavators using proprioceptive force sensing and machine learning. Autom Constr. 2020. https://doi.org/10.1016/j.autcon.2020.103374.
37. Faubel L, Schmid K, Eichelberger H. Is MLOps different in Industry 4.0? General and specific challenges. In: Proceedings of the 3rd International Conference on Innovative Intelligent Industrial Production and Logistics—IN4PL. SciTePress, Valletta, Malta, 2022; pp 161–167. https://doi.org/10.5220/0011589600003329.
38. Sato D, Wider A, Windheuser C. Continuous delivery for machine learning. visited 2022-06-18 (2019). https://martinfowler.com/articles/cd4ml.html.
39. Hannelius T, Salmenpera M, Kuikka S. Roadmap to adopting OPC UA. In: 2008 6th IEEE International Conference on Industrial Informatics, 2008; pp 756–761. https://doi.org/10.1109/INDIN.2008.4618203.
40. Iñigo MA, Porto A, Kremer B, Perez A, Larrinaga F, Cuenca J. Towards an asset administration shell scenario: a use case for interoperability and standardization in industry 4.0. In: Network Operations and Management Symposium, 2020; pp 1–6. https://doi.org/10.1109/NOMS47738.2020.9110410.
41. Ruf P, Reich C, Ould-Abdeslam D. Aspects of module placement in machine learning operations for cyber physical systems. In: 2022 11th Mediterranean Conference on Embedded Computing (MECO), 2022; pp 1–6. https://doi.org/10.1109/MECO55406.2022.9797080.
42. Fujii TY, Hayashi VT, Arakaki R, Ruggiero WV, Bulla R, Hayashi FH, Khalil KA. A digital twin architecture model applied with mlops techniques to improve short-term energy consumption prediction. Machines. 2022. https://doi.org/10.3390/machines10010023.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.