Towards MLOps: A Framework and Maturity Model

Conference Paper · September 2021

DOI: 10.1109/SEAA53835.2021.00050

3 authors:

Meenu mary John
Malmö University

Helena Holmstrom Olsson
Malmö University

Jan Bosch
Chalmers University of Technology

Towards MLOps: A Framework and Maturity Model

Meenu Mary John
Computer Science and Media Technology
Malmö University
Malmö, Sweden
meenu-mary.john@mau.se

Helena Holmström Olsson
Computer Science and Media Technology
Malmö University
Malmö, Sweden
helena.holmstrom.olsson@mau.se

Jan Bosch
Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden
jan.bosch@chalmers.se

***Abstract*—The adoption of continuous software engineering practices such as DevOps (Development and Operations) in business operations has contributed to significantly shorter software development and deployment cycles. Recently, the term MLOps (Machine Learning Operations) has gained increasing interest as a practice that brings together data scientists and operations teams. However, the adoption of MLOps in practice is still in its infancy and there are few common guidelines on how to effectively integrate it into existing software development practices. In this paper, we conduct a systematic literature review and a grey literature review to derive a framework that identifies the activities involved in the adoption of MLOps and the stages in which companies evolve as they become more mature and advanced. We validate this framework in three case companies and show how they have managed to adopt and integrate MLOps in their large-scale software development companies. The contribution of this paper is threefold. First, we review contemporary literature to provide an overview of the state-of-the-art in MLOps. Based on this review, we derive an MLOps framework that details the activities involved in the continuous development of machine learning models. Second, we present a maturity model in which we outline the different stages that companies go through in evolving their MLOps practices. Third, we validate our framework in three embedded systems case companies and map the companies to the stages in the maturity model.**

***Index Terms*—MLOps, Framework, Maturity Model, SLR, GLR, Validation Study**

I. Introduction

Machine Learning (ML) has a significant impact on the decision-making process in companies. As a result, companies can save significant costs in the long run while ensuring value for their customers [1] and also enabling fundamentally new ways of doing business. To improve value creation and automate the end-to-end life cycle of ML, data scientists and operations teams are trying to apply DevOps concepts to their ML systems [2] in companies. DevOps is a "set of practices and tools focused on software and systems engineering" [3] with close collaboration between developers and operations teams to improve quality of service [4]. ML models embedded in a larger software system [5] are only a small part of the overall software system, so the interaction between the model and the rest of the software and its context is essential [6]. From literature it is apparent that ML processes are often not well integrated with continuous development and production in practice [2].

Software Center

Despite the popularity of ML, there is little research on MLOps because it is a recent phenomenon. To advance understanding of how companies practice MLOps, including collaboration between data science and operations teams, we use a Systematic Literature Review (SLR), a Grey Literature Review (GLR), and a validation study in three case companies. The paper makes three contributions.

- We conduct a SLR and a GLR literature review to present the state-of-the-art regarding the adoption of MLOps in practice and derive a framework from the reviews
- We present a maturity model with different stages in which companies evolve during MLOps adoption
- We validate the framework and map the three case companies to the stages of the maturity model

The remainder of the paper is organized as follows: Section II describes the background of the study, Section III describes the research methods used and Section IV addresses the threats to validity. Section V summarizes the findings from the literature review. Section VI describes the MLOps framework and maturity model. Section VII describes the validation study conducted in three case companies and Section VIII discusses the results. Section IX concludes our study.

II. Background

This section discusses DevOps, DevOps application on the ML systems (referred to as MLOps), and the challenges associated with it.

A. DevOps

DevOps [3] aims to "reduce the time between committing a change to a system and the change being placed into normal production while ensuring high quality" [7]. The goal is to merge development, quality assurance, and operations into a single continuous process. The key principles of DevOps are automation, continuous delivery, and rapid feedback. DevOps requires a "delivery cycle that involves planning, development, testing, deployment, release and monitoring as well as active cooperation between different team members" [3].

Continuous software engineering (SE) refers to iterative software development and related aspects like continuous integration, continuous delivery, continuous testing and continuous deployment. Continuous SE enables development, deployment and feedback at a rapid pace [8] [9] and is divided into three phases: a) Business strategy and planning, b) Development and c) Operations. Software development activities such as continuous integration (CI) and continuous delivery (CD) support the operations phase. In CI [8], team members of software-intensive companies often integrate and merge development code to have a faster and more efficient delivery cycle and increases team productivity [9]. This facilitates the automation of software development and testing [10]. CD ensures that an application is not moved to the production phase until automated testing and quality checks have been successfully completed [11] [12]. It lowers deployment risk, cost, and provides rapid feedback to users [13] [14].

B. MLOps

With the successful adoption of DevOps, companies are looking for continuous practices in the development of ML systems. To unify the development and operation of ML systems, MLOps [5] extends DevOps principles [15]. In addition to traditional unit and integration testing, CI introduces additional testing procedures such as data and model validation. From the perspective of CD, processed datasets and trained models are automatically and continuously delivered by data scientists to ML systems engineers. From the perspective of continuous training (CT), introduction of new data and model performance degradation require a trigger to retrain the model or improve model performance through online methods. In addition, appropriate monitoring facilities ensure proper execution of operations.

C. Challenges associated with MLOps

In our own previous research [16] [17], we have identified a number of challenges when it comes to the business case, data, modeling and deployment of ML or Deep Learning (DL) models. These include high AI costs and expectations, fewer data scientists, need for large datasets, privacy concerns and noisy data, lack of domain experts, labeling issues, increasing feature complexity, improper feature selection, introduction of bias when experimenting with models, highly complex DL models, need for deep DL knowledge, difficulty in determining final model, model execution environment, more hyperparameter settings, and verification and validation. It also includes less DL deployment, integration issues, internal deployment, need for an understandable model, training-serving skew, end-user communication, model drifts, and maintaining robustness. Some of the challenges in MLOps practice [5] include tracking and comparing experiments, lack of version control, difficulty in deploying models, insufficient purchasing budgets and a challenging regulatory environment.

III. Research Method

The main objective of the study is to identify the activities associated with the adoption of MLOps and the stages in which companies evolve as they gain maturity and become more advanced. To achieve this objective, we developed the following research questions:

- RQ1: What is the state-of-the-art regarding the adoption of MLOps in practice and the different stages that companies go through in evolving their MLOps practices?
- RQ2: How do case companies evolve and advance their MLOps practices?

We performed a SLR [18] [19], a GLR [20] [21] and a validation case study [22] to address the two RQs.

A. SLR and GLR

The goal of the SLR is to find, examine and interpret relevant studies on the topic of interest [18] [19]. To answer the RQs, we defined search strings according to [18] and searched five popular scientific libraries. Figure 1 shows an overview of the SLR and the GLR process that was used in this study. We integrated and exported relevant studies into an Excel spreadsheet for deeper analysis. In SLR, we included conference and journal studies that reported MLOps. On the other hand, we excluded studies that were duplicate versions, published in a language other than English, were not peer-reviewed, and were not available electronically on the Internet.

We conducted the GLR [20] to provide a detailed description of the state-of-practice and practitioner experiences in adopting MLOps. Compared to the SLR, the GLR provides the voice of practitioners on the topic under study. In GLR, we included studies in the Google Search that address MLOps, published in English in PDF format and documents from companies by filtering the site under the domain name ".com". To improve the reliability of the retrieved results from the GLR, we excluded peer-reviewed scientific articles and other sources of knowledge such as blogs, posts, etc.

For the SLR and the GLR, we used the search query as "MLOps" OR "Machine Learning Operations" and restricted the search to the period between January 1, 2015 and March 31, 2021. The time interval was chosen because the term MLOps is prevalent after the concept "Hidden Technical Debt in Machine Learning Systems" [6] in 2015. Based on the SLR and the GLR, we shortlisted 6 SLR ([23] - [28]) and 15 GLR [29] - [43] studies. Based on these studies, we developed an MLOps framework and various stages that companies take in evolving MLOps practices.

B. Validation Case Study

Following [44], we conducted a validation study to map companies to the stages in the maturity model derived from literature reviews. Case study methodology is an empirical research approach based on an in-depth study of a contemporary phenomenon that is difficult to study separately in its real-world environment [45]. In SE, case studies are used to better understand how and why SE was done and thus improve the SE process and resulting software products [46]. Throughout the validation study, we worked closely with practitioners in each case company. Table 1 provides a brief description of each case company, the practitioners (P*,

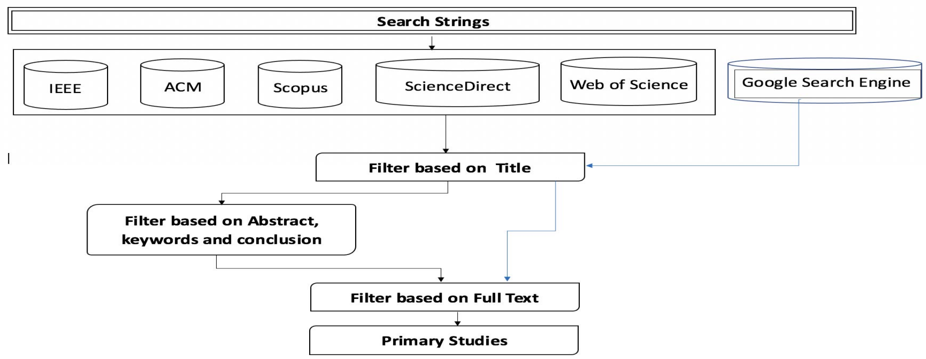

Fig. 1. Overall SLR and GLR process used in the study

W*, M*, and S* represent interview, workshop, meeting, and stand-up meeting participants respectively) and their roles.

TABLE I
DESCRIPTION OF PRACTITIONERS IN THE VALIDATION STUDY

Case Company	Practitioners	
	ID	***Roles***
Telecommunications	P1, W1, S1	Senior Data Engineer
	P2, W2, S2	Data Scientist
	P3, W3, S3	Data Scientist
	P4, W4, S4	Data Scientist
	W5, S5	Senior Data Scientist
	W6, S6	Data Scientist
	W7, S7	Software Developer
	W8, S8	Software Developer
	S9	Operational Product Owner
	S10	Sales Director
Automotive	W9	Expert Engineer
	W10	Expert Engineer
Packaging	M1	Solution Architect
	M2	Data scientist

Case Companies: We present three case companies and use cases that were investigated in each company as part of our validation study.
1. Hardware Screening: The telecommunications company predicts faults in hardware to minimize the amount of hardware returned by the customer for repair. In this use case, they focus on a) Returning defect-free hardware back to the customer b) Sending defective hardware to the repair center.
2. Self-driving Vehicles: The company that manufactures vehicles strives to provide autonomous transportation solutions. The main use case is self-driving vehicles to increase the productivity. The company also needs to ensure that the failure rate is low in these safety-critical use case.
3. Defect Detection: The packaging company provides packaging solutions as well as machines to customers. One of the main use cases is the detection of defects in finished/semi-finished packages.

Data Collection and Analysis: For data collection, we used interview studies, workshops, meetings and stand-up meetings in companies. They were held in English via video conferencing. All interviews lasted 45 minutes, workshops and meetings lasted 30 minutes to one hour, and daily stand-up meetings lasted 15 minutes. We validated the MLOps framework in case companies and present different stages that companies go through when implementing MLOps. Transcripts from interviews and notes from workshops, meetings and stand-ups were used to capture empirical data. Later, they were shared with the other authors by primary author for detailed analysis. We applied elements of open coding to analyse and categorize collected empirical data [47]. In order to obtain different perspectives on the topic under study, triangulation was used [48].

IV. THREATS TO VALIDITY

Potential validity threats were considered and minimized in this study [49]. Construct validity was improved by considering information from SLR, GLR and the validation case study. Authors and practitioners involved in this study are well versed in MLOps. Multiple techniques (semi-structured interviews, workshops, meetings, and stand-up meetings) and multiple sources (senior data engineer, data scientist, software developer, expert engineer, etc.) were used to collect and validate empirical data. Internal validity threats caused by faulty conclusions due to primary author bias in data selection or interpretation are mitigated by consulting with other two authors. By extending our research to additional case companies, generalization of the results can be justified and thus external validity can be mitigated.

V. LITERATURE REVIEW FINDINGS

Based on the SLR and the GLR, we extract insights from the literature to give an overview of the state-of-the-art of MLOps in practice. They are divided into three parts: a) Data for ML Development b) ML Model Development and c) Release of ML Models. Below, we discuss each part in detail.

A. Data for ML Development

Aggregate heterogeneous data from different data sources [32] [31] [41], preprocessing [27] and extracting relevant features are necessary to provide data for ML development. Later, the features are registered in a feature store [42] which can be used for development of any ML models [42] and used for inference when deploying the model. Also, the data points are stored in the data repository [39] after versioning. The data collected from various sources has to be properly stored and managed. Data anonymization and encryption [27] should be performed to comply with data regulations (e.g. GDPR [25]).

B. *ML Model Development*

In ML model development, provisions should be made to run experiments in parallel, optimize the chosen model with hyperparameters, and finally evaluate the model to ensure that it fits the business case. After versioning, the code is stored in the code repository [42] [23]. The model repository [39] keeps track of the models that will be used in production, and the metadata repository contains all the information about the models (e.g., hyperparameter information). Data scientists can collaborate on the same code base, which also allows them to run the code in different environments and against a variety of datasets. This facilitates scaling and the ability to track the execution of multiple experiments and reproducibility [29].

C. *Release of ML Models*

To release ML models, package [41], validate [41] and deploy models [40] to production [41]. When deploying a model to production, it has to be integrated with other models as well as existing applications [30] [41]. When the model is in production, it serves requests. Despite the fact that training is often a batch process, the inferences can be REST endpoint/custom code, streaming engine, micro-batch, etc. [35]. When performance drops, monitor the model [41] and enable the data feedback loop [41] to retrain the models . In a fully mature MLOps context, perform continuous integration and delivery by enabling the CI/CD pipeline and continuous retraining through CT pipeline [41] [31].

From the literature review, we see that successful AI/ML operationalization ensures a safe, traceable, testable, and repeatable path for developing, training, deploying, and updating ML models in different environments [30]. The use of MLOps enables automation, versioning, reproducibility, etc., with successful collaboration of required skills such as data engineer, data scientist, ML engineer/developer [40] [29]. For example, data scientists must specialize in SE skills such as modularization, testing, versioning, etc. [36]. Supporting processes formalized in policies serve as the basis for governance [31] and can be automated to ensure solution reliability and compliance [31]. MLOps also support explainability (GDPR regulation [25]) and audit trails [40].

VI. MLOps Framework and Maturity Model

Based on the SLR and the GLR, we derive an MLOps framework that identifies the activities involved in MLOps adoption. Figure 2 depicts the MLOps framework. The entire framework is divided into three pipelines: a) Data Pipeline b) Modeling Pipeline and c) Release Pipeline. After collecting data relevant to ML models from data sources, preprocessing of data and feature extraction is performed. Once a suitable model has been experimented and optimized with hyperparameters, evaluate the model and package it for production deployment. If performance degrades, trigger retraining of the model by initiating a data feedback loop. Data and code that have been versioned are stored in the data repository and code repository. To track the deployable model version, store it in the model registry. Deployment cycles of ML models can be shortened using CI/CD/CT.

MLOps Maturity Model: Based on the SLR and the GLR, we present a maturity model in which we outline four stages in which companies evolve when adopting MLOps practices. The four stages are a) Automated Data Collection b) Automated Model Deployment c) Semi-automated Model Monitoring and d) Fully-automated Model Monitoring. These stages capture key transition points in the adoption of MLOps in practice. Below, we detail each MLOps stage and preconditions for a company to reach this stage.

A. Automated Data Collection: In this stage, companies have a manual processing of data, model, deployment and monitoring. With the adoption of MLOps, company experience a transition from manual process to automated data collection for (re)training process.

Preconditions: For transition from manual process to automated data collection, there is a need for mechanism to aggregate data from different data sources which can be stored and accessed whenever required [32]. In addition, it also demands capability for integrating and processing new data sources, regardless of variety, volume or velocity [31]. It also requires infrastructure resources for automated data collection [34], data preparation and collaboration [38]. Also, standardized and automated pipelines helps to drive ingestion, transformation and storage of analytic data into a database or data lake [31]. Same feature manipulation during training has to be replicated at the inference [35]. AI teams can promote trust by addressing data management challenges like accountability, transparency, regulation and compliance, and ethics [37].

B. Automated Model Deployment: The companies at this stage have a manual model deployment and monitoring. With the adoption of MLOps, they undergo transition from manual model deployment and monitoring to automated deployment of the retrained model.

Preconditions: The transition can be achieved by implementing provisions for automated model deployment to environments [43] [39] [43] [38] especially across dev, Q/A and production environments [43] [34]. It encourages deployment freedom in on-premise, cloud and edge [34] [38]. Automated deployment of retrained model can be achieved by providing a dedicated infrastructure-centric CI/CD pipeline [31], integration with DevOps for automation, scale and collaboration [35]. Sufficient infrastructure choices for deployment includes model hosting, evaluation, and maintenance [32], and means to register, package (containerization [38] [24]), deploy models [43] [40] [39] and integration of reusable software environments for training and deploying models [43]. Tracking experiments [43] [39] [40] and models [31],

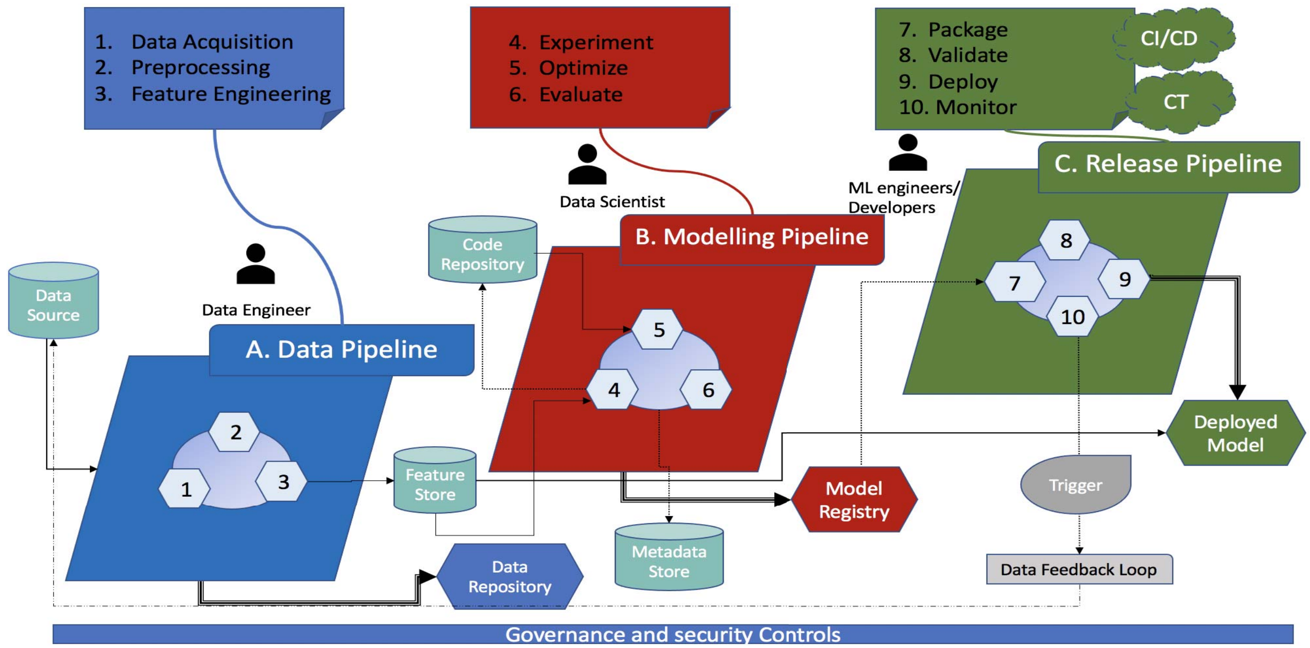

Fig. 2. MLOps Framework

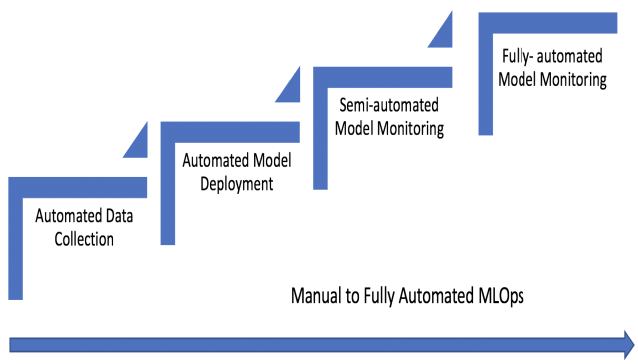

Fig. 3. MLOps Maturity Model

proper validation of models and data [36] can accelerate automated model deployment. Canary Deployments [32] and provisions to store, annotate, discover, and manage models in a central repository [39] can facilitate automated deployment of retrained model. This stage also requires multi-talented teams of technologists and ML professionals to operationalize and scale AI [37].

C. Semi-automated Model Monitoring: At this stage, companies have a manual model monitoring in place. With MLOps, they can attain a transition from manual monitoring to semi-automated model monitoring.

Preconditions: To reach this transition, there should be provisions for triggering [43] when performance degrades and availability of tools for diagnostics, performance monitoring and addressing model drift [43] [27] [36]. It also requires automation scripts to manage and monitor models based on drift [38] and ability to perform continuous model tracking [31]. For easy monitoring of models, MLOps professionals has to be provided with visual tools [34], and dedicated and centralized dashboards [38] [27] [28]. It also requires data orchestration pipelines and rule-based data governance to ensure data changes [31], feedback loop and continuous model retraining [43]. There should be also a mechanism to automatically train model in production using fresh data based on live pipeline triggers and feedback loops [38]

D. Fully-automated Model Monitoring: The companies have deployment and monitoring of models in place where performance degradation is acknowledged by alert. By utilizing MLOps, they undergo transition towards fully automated monitoring of models.

Preconditions: For this transition, company requires CI/CD integration with automation and orchestration [43] and CT pipeline to retrain models when performance degrades [31]. For this transition, there is a need to ensure certification of models [32] [23], governance and security controls [43] [34] [36], model explainability [43] [36], auditing of model usage [34] [43], reproducible workflow and models [36]. There should be mechanisms to perform end-to-end QA test and performance checks [43]. There should be assurance that data security and privacy requirements are built into data pipelines [31] as well as retrain production models on newer data using the data, algorithms and code used to create the original [34].

VII. Validation Study

The framework derived from the previous literature was validated in three case companies. Below, we detail how they have tried to introduce and integrate MLOps into their software development systems.

A. Case Company A

Prior to implementing MLOps, the company planned an initial meeting with team members to discuss realistic expectations for MLOps. According to *P2*, practitioners must spend a significant amount of time creating the architecture, communicating, and discussing MLOps in the beginning, the end result is a significant reduction in manual work and end-to-end automation. According to case company A, the primary goal of MLOps is to achieve a) Automation b) Versioning of datasets and models c) Traceability and d) Reproducibility.

> P1: *"We have already implemented something, but not fully, we are still investigating the concept of MLOps. We are trying to see what is in place and what is not and also how to implement that are not in place"*

When working with data, practitioners need to have a data pipeline in place and provisions to register training data. Before MLOps was implemented, this process was manual in the company. For instance, when practitioners train a model, they read data from log files and ended up using new data each time to train because they did not have access to old data due to the lack of a data pipeline. To ensure the quality of the data pipeline, data schema has to be validated. The company is thinking about using DVC (Data Version Control) to facilitate comparison of different models and visualizations. When dealing with models, practitioners keep track of model performance by tuning them with hyperparameters and maintain the quality of the model pipeline.

Practitioners in case company A place more emphasis on understanding how a project works, especially when it comes to the concept of model deployment. According to P3, the best way to verify that integration of ML code into a project works or not is to have a CI or mechanism to test it without running it in production. For example, practitioners develop a program that can run on a laptop and give everyone access to a local repository and environment. In such a way they can run the data and the entire project with less data set (for instance, 10 percent of the total size of the data set) to make sure it works properly. If the models work properly in the development environment, practitioners will move them to the production environment. To integrate these models, they need to consider other models that are already in production. As a result, it takes time for practitioners to understand how to feed previous data into the system. The company uses Tableau for data visualization and Graffana for model monitoring.

In case company A, data pipeline is quite immature and the model pipeline is not fully automated. On the other hand, model serving pipeline is quite aligned with MLOps. The company utilizes dataset versioning to compare models. They earlier uses Gate for versioning and recently moved to artifactory.

> W5: *"We have versioning for data and model but may be in future we will go for versioning up the entire pipeline, configurations etc"*
>
> P1: *"There are somethings which we do in our day to day work which are continuous and some of these can be automatized. A data scientist should focus on feature engineering, model training etc., instead of deployment or even writing test cases which can be automatized to a great extent"*

B. Case company B

Case company B is also trying to implement MLOps in its context. In dealing with data, the company collects data from real vehicles or generates it using simulations. They input this data into a logging system and add metadata to the logs. After labeling, they ensure the quality of the images. Practitioners access the logged data via an API, which is offloaded from hard drives to servers. To extract data, they perform property selection on selected data frames, run queries to find data to select for future investigation, run algorithms to find valuable data and anonymize data. The company also looks forward to ensuring consistency of annotations throughout the project. Once the dataset is annotated, they perform pre-processing and split the dataset into training, validation, and testing.

When it comes to models, they train neural networks, spin computational nodes, and deallocate them after training. They back up the experiments and validate the models and use model pruning to increase inference speed. The practitioners convert the model using ONXX and deploy it to artifactory. Once the model is deployed in Artifactory, it can be used in vehicles. They essentially deploy the pipeline using the CI/CD loop. They also update the validation set based on new data, domain, etc. The company is interested in moving from on-premise to cloud services for scalability. The company is creating artifacts.

> W8: *"We build artifacts as we need tool chain for data selection, development and deployment on target devices to run inferences. We depend on other teams inside the company for certain artifacts. For instance, the logging system. Besides that, the team build the rest of the artifacts."*

C. Case company C

Case company C drive towards attaining fully automated MLOps. Company has standardized and quality model development, manageable deployment workflow and model lifecycle in production. The main architectural principles of case company C is to achieve a) Scale and high availability b) Flexibility and extensibility c) Integration d) Automation and e) Maintainability.

When dealing with data, the company captures images of packages using cameras or generate them using simulations. The data captured is stored in data lake before using it for training. The practitioners experiment with several algorithms before finalizing the best suitable model and adopt hyper-parameter tuning. They utilize GPUs for training to reduce needed time. The company applies DevOps principles to their ML systems. The models are packaged and deployed to production via docker containers. They employ Kubernetes to automate deployment as well as scaling. The company has provisions for tracking data, models and experiments. They have model management in place and also models can be deployed to cloud, edge and on-premise. The model monitoring can be visualized using dashboards and retrain models when performance degrades. The company also uses tool chain for development and deployment of models.

VIII. Discussion

This study highlights the emerging interest in MLOps and the increasing adoption of these practices in software-intensive systems. Compared to SLRs ([23] - [28]), more relevant GLR studies [29] - [43] on MLOps are retrieved from the literature. This is a positive sign as it gives an indication that more companies are driving towards achieving fully automated MLOps. Both the SLR and the GLR emphasize the fact that cross-functional teams with skills from data engineering, data science, or operations can facilitate MLOps. Based on insights from the literature, we see that feature store, data repository, code repository, metadata store, model registry, and feed-back loops can shorten the transition of models from prototype to production stage. As a result, they promote automation, versioning, explainability, and traceability.

In Figure 4, we see that case company A is placed in between the phases - Automated Data Collection and Automated Model deployment. This is because while the modeling and deployment pipelines in this company are very mature, the data pipeline is immature. Also the company intends to version the data, model and release pipelines. The challenges faced by case company A in the beginning phases of MLOps corresponds to challenges we identified in literature reviews. The data pipeline challenges that company A is experiencing is something common and also reported by other companies in studies that were part of the GLR. Similar to case company A, we place company B at stage one in the maturity model - Automated Data Collection. This is because they are looking forward to ensuring consistency of annotations across project. On the other hand, they have provisions for data collection from multiple sources, queries and algorithms to run valuable data, experiment tracking, etc. Since their data pipeline is not completely automated, we place them in stage one. Case company C is placed in stage two - Automated Model deployment in the maturity model. They employ DevOps principles in their application, data lakes available for collecting data, frameworks for running experiments, track experiments, utilize docker containers for deployment. They also have mechanisms to deploy models in cloud or edge. Whenever degrades, they initiate model retraining and has model management in place. Even though they have mechanisms for automated model deployment, they are placed in stage two as they look forward to achieve fairness, generalizability, explainability and governance of models.

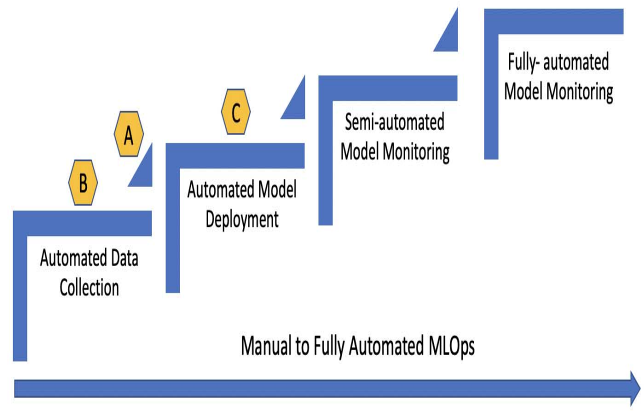

Fig. 4. Mapping: Case companies to stages in Maturity model

IX. Conclusion

Companies adopt DevOps principles to ML systems in order to allow continuous development, deployment and delivery of these systems. In this paper, we derive a framework that identifies the activities involved when adopting MLOps and the stages in which companies evolve as they become more advanced. We validate this framework in three software-intensive embedded systems companies and highlight how they have managed to adopt and integrate MLOps into their large-scale software development organizations. In future research, we plan to expand our study by involving additional case companies and experts for validation of our results. We believe our findings support successful adoption of MLOps in software-intensive embedded systems companies.

X. Acknowledgment

We would like to thank all case companies and practitioners involved in the validation study for sharing their experiences and examples for this paper. This work is funded by Software Center.

References

[1] Miklosik, A., Kuchta, M., Evans, N. and Zak, S., 2019. Towards the adoption of machine learning-based analytical tools in digital marketing. IEEE Access, 7, pp.85705-85718.
[2] Karamitsos, I., Albarhami, S. and Apostolopoulos, C., 2020. Applying DevOps practices of continuous automation for machine learning. Information, 11(7), p.363.
[3] DevOps Documentation. Available online: https://devops.com (accessed on 25 March 2021).
[4] Farroha, B.S.; Farroha, D.L. A framework for managing mission needs, compliance, and trust in the DevOps environment. In Proceedings of the IEEE Military Communications Conference, Baltimore, MD, USA, 6–8 October 2014; pp. 288–293.
[5] Mäkinen, S., Skogström, H., Laaksonen, E. and Mikkonen, T., 2021. Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?. arXiv preprint arXiv:2103.08942.
[6] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F. and Dennison, D., 2015. Hidden technical debt in machine learning systems. Advances in neural information processing systems, 28, pp.2503-2511.
[7] Zhu, L., Bass, L. and Champlin-Scharff, G., 2016. DevOps and its practices. IEEE Software, 33(3), pp.32-34

[8] Fitzgerald, B.; Stol, K.J. Continuous Software Engineering: A Roadmap and Agenda. J. Syst. Softw. 2017, 123, 176–189.
[9] Bosch, J. Continuous Software Engineering: An Introduction. In Continuous Software Engineering; Springer: Berlin/Heidelberg, Germany, 2014; pp. 3–13
[10] Leppanen, M.; Makinen, S.; Pagels, M.; Eloranta, V.P.; Itkonen, J.; Mantyla, M.V.; Mannisto, T. The Highways and Country Roads to Continuous Deployment. IEEE Softw. 2015, 32, 64–72
[11] Weber, I.; Nepal, S.; Zhu, L. Developing Dependable and Secure Cloud Applications. IEEE Internet Comput. 2016, 20, 74–79.
[12] Humble, J. Continuous Delivery vs. Continuous Deployment. Available online: https://continuousdelivery. com/2010/08/continuous-delivery-vs-continuous-deployment (accessed on 25 March 2021).
[13] Chen, L. Continuous Delivery: Huge Benefits, but Challenges Too. IEEE Softw. 2015, 32, 50–54.
[14] Humble, J. What is Continuous Delivery? Available online: https://continuousdelivery.com/2010/02/ continuous-delivery/ (accessed on 25 March 2021).
[15] Google Cloud. 2020. MLOps: Continuous delivery and automation pipelines in machine learning. Retrieved from https://cloud.google.com/solutions/machine-learning/mlops-continuousdelivery-and-automation-pipelines-in-machine-learning.
[16] John, M.M., Olsson, H.H. and Bosch, J., 2020, June. Developing ML/DL models: A design framework. In Proceedings of the International Conference on Software and System Processes (pp. 1-10).
[17] John, M.M., Olsson, H.H. and Bosch, J., 2020, August. AI on the Edge: Architectural Alternatives. In 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 21-28). IEEE.
[18] Keele, S., 2007. Guidelines for performing systematic literature reviews in software engineering (Vol. 5). Technical report, Ver. 2.3 EBSE Technical Report. EBSE.
[19] Kitchenham, B.A., Budgen, D. and Brereton, P., 2015. Evidence-based software engineering and systematic reviews (Vol. 4). CRC press.
[20] Garousi, V., Felderer, M. and Mäntylä, M.V., 2016, June. The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In Proceedings of the 20th international conference on evaluation and assessment in software engineering (pp. 1-6).
[21] Williams, A., 2018, June. Using reasoning markers to select the more rigorous software practitioners' online content when searching for grey literature. In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 (pp. 46-56).
[22] Yin, R.K., 2017. Case study research and applications: Design and methods. Sage publications.
[23] Zhou, Y., Yu, Y. and Ding, B., 2020, October. Towards MLOps: A Case Study of ML Pipeline Platform. In 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE) (pp. 494-500). IEEE.
[24] Silva, L.C., Zagatti, F.R., Sette, B.S., dos Santos Silva, L.N., Lucrédio, D., Silva, D.F. and de Medeiros Caseli, H., 2020, December. Benchmarking Machine Learning Solutions in Production. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 626-633). IEEE.
[25] Tamburri, D.A., 2020, September. Sustainable MLOps: Trends and Challenges. In 2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) (pp. 17-23). IEEE.
[26] Vuppalapati, C., Ilapakurti, A., Chillara, K., Kedari, S. and Mamidi, V., 2020, December. Automating Tiny ML Intelligent Sensors DevOPS Using Microsoft Azure. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 2375-2384). IEEE. Banerjee, A., Chen, C.C., Hung, C.C.,
[27] Huang, X., Wang, Y. and Chevesaran, R., 2020. Challenges and experiences with mlops for performance diagnostics in hybrid-cloud enterprise software deployments. In 2020 USENIX Conference on Operational Machine Learning (OpML 20).
[28] Lim, J., Lee, H., Won, Y. and Yeon, H., 2019. MLOP lifecycle scheme for vision-based inspection process in manufacturing. In 2019 USENIX Conference on Operational Machine Learning (OpML 19) (pp. 9-11).
[29] MLOps: Continuous Delivery for Machine Learning on AWS. https://d1.awsstatic.com/whitepapers/mlops-continuous-delivery-machine-learning-on-aws.pdf
[30] MLOps Foundations. https://www.rackspace.com/sites/default/files/white-papers/MLOps_Foundations_Data_Sheet.pdf
[31] MLOps. https://perspecta.com/sites/default/files/2021-02/MLOps%20Whitepaper_022321.pdf
[32] Machine Learning Operations (MLOps) A Meetup. https://higherlogicdownload.s3.amazonaws.com/IMWUC/UploadedImages /12f13a33-bced-4573-8dd7-be58d519757c/MLOps.pdf
[33] Inroducing MLOps. https://itlligenze.com/uploads/5/137039/files/oreilly-ml-ops.pdf
[34] Eight must-haves for MLOps success and when to use them. https://info.algorithmia.com/hubfs/2020/Webinars/Forrester/Eight-must-haves-slides-final.pdf?hsLang=en-u
[35] MLOps: Machine Learning Operationalization. https://cdn.activestate.com/wp-content/uploads/2018/10/webinar-slides-mlops.pdf
[36] What is MLOps. https://blogs.bmc.com/mlops-machine-learning-ops/?print=pdf
[37] MLOps: Industrialized AI. https://www2.deloitte.com/content/dam/insights /articles/7022_TT-MLOps-industrialized-AI/DI_2021-TT-MLOps_infographic.pdf
[38] Deolite. https://www2.deloitte.com/content/dam/Deloitte/us/Documents /technology/us-ml-oops-to-mlops.pdf
[39] MLOps using MLFlow. https://www.iteblog.com/ppt/dataai-summit-europe-2020/mlops-using-mlflow-iteblog.com.pdf
[40] Seamless MLOps with Seldon and MLflow. https://www.plainconcepts.com/wp-content/uploads/2020/12/Seamless-MLOps-with-Seldon-and-MLflow.pdf
[41] MLOps with Azure AutoML. https://query.prod.cms.rt.microsoft.com /cms/api/am/binary/RE4KW7y
[42] MLOps stack. https://valohai.com/assets/files/mlops-stack.pdf
[43] Onica. https://insights.onica.com/hubfs/2020%20Offerings/2020 _Onica_MLOps_Foundations_Offer.pdf
[44] Gerring, J., 2004. What is a case study and what is it good for?. American political science review, pp.341-354.
[45] Walsham, G., 1995. Interpretive case studies in IS research: nature and method. European Journal of information systems, 4(2), pp.74-81.
[46] Runeson, P., Host, M., Rainer, A. and Regnell, B., 2012. Case study research in software engineering: Guidelines and examples. John Wiley & Sons.
[47] Holton, J.A., 2007. The coding process and its challenges. The Sage handbook of grounded theory, 3, pp.265-289.
[48] Wilson, V., 2014. Research methods: triangulation. Evidence based library and information practice, 9(1), pp.74-75.
[49] Easterbrook, S., Singer, J., Storey, M.A. and Damian, D., 2008. Selecting empirical methods for software engineering research. In Guide to advanced empirical software engineering (pp. 285-311). Springer, London.
[50] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.