A Maturity Level Framework for Practicing Machine Learning Operations in CI/CD For Software Deployment

Muhammad Adeel Mannan  
Department of Computing, Faculty of Engineering Sciences and Technology (FEST), Hamdard University, Karachi, Pakistan  
adeel.mannan@hamdard.edu.pk

Sumaira Mustafa  
Department of Computing, Faculty of Engineering Sciences and Technology (FEST), Hamdard University, Karachi, Pakistan  
sumaira.mustafa@hamdard.edu.pk

Afzal Hussain  
Department of Computing, Faculty of Engineering Sciences and Technology (FEST), Hamdard University, Karachi, Pakistan  
afzal.hussain@hamdard.edu.pk

Abstract – Significantly shorter software development and deployment cycles have been made possible by the adoption of continuous software engineering techniques in business operations, such as DevOps (Development and Operations). Data scientists and operations teams have recently become more and more interested in a practice known as MLO (Machine Learning Operations). However, MLO adoption in practice is still in its early stages, and there aren't many established best practices for integrating it into current software development methods. In order to give a frame that outlines the way necessary in espousing MLO and the stages through which business capes process as they come riper and sophisticated, we achieve a methodical literature study as well as a slate review of literature in this composition. We test this approach in three example businesses and demonstrate how they were able to embrace and incorporate MLO into their massive software development businesses. This study offers three contributions. To describe the current level of knowledge on the use of MLO in practice, we undertake a literature study. From the reviews, we build a framework.

Index Terms – Software DevOps, Machine Learning, Maturity Model, Validation Study, CI/CD, Framework

I. INTRODUCTION

The way decisions are made in businesses is significantly impacted by machine learning (ML). As a consequence, businesses may make long-term cost savings that guarantee value for their clients [1] and also open the door to radically new business models. Data scientists and operations staff are attempting to apply DevOps ideas to their ML systems in organizations in order to enhance value generation and automate the form start to finish life cycle of ML [2]. A "set of practices and tools that concentrate on systems and software engineering"

(DevOps) [3] involves developers and operations teams working closely together to enhance service quality [4]. Because ML models are only a small portion of a larger software system, the relationship between the model itself and the remainder of the software as well as its context is crucial [6]. It is clear from the literature that in practice, ML procedures are frequently not adequately linked with continuous development and production [2].

Despite the widespread usage of ML, little study has been done on MLO because it is a relatively new topic. We employ a systematic review of the literature (SLR), a grey literature research (GLR), and a validation investigation in three case firms to further our knowledge of how organizations practice MLO, including collaboration between data science and operations teams.

Three contributions are made by the paper. To describe the current level of knowledge on the use of MLO in practice, we undertake a literature study. From the reviews, we build a framework. The study's background is covered in Section 2, the re-search methodologies are covered in Section 3, and validity risks are covered in Section 4.

The conclusions drawn from the research review are compiled in Section 5. The MLO structure and maturity model are described in Section 6. Section 8 addresses the findings of Section 7 summarize the validation research that was carried out in three case firms. Our investigation is concluded in Section 9.
II. BACKGROUND

This section examines DevOps, its implementation to ML systems (known as MLO), and the difficulties it faces.

A. DevOps

DevOps [3] was created to “reduce the time between negotiating a remodeling to a system and the remodeling being placed into common output while icing high quality” [7]. The objective is to combine operations, quality control, and development into a single continuous process. In DevOps, automation, continuous delivery, and quick feedback are the guiding principles [3] necessary for DevOps.

Iterative software development and associated concepts like continuous integration, continuous delivery, continuous testing, and continuous deployment are referred to as continuous software engineering (SE) [5-7]. Continuous integration (CI) and continuous delivery (CD) are examples of software development processes that assist the operations phase. Software-intensive firms frequently integrate and merge development code as part of CI [8], which speeds up the delivery cycle and boosts team productivity [9]. This makes software development and testing more automated [10]. CD makes ensuring that an application is not put into production before automated testing and quality checks have been successfully performed [11] [12]. It reduces deployment risk, costs, and gives consumers quick feedback [13][14].

B. MLO

MLO [5] advances DevOps ideas [15] to harmonize the creation and maintenance of ML systems. Traditional unit and integration testing are supplemented by additional testing techniques like data and model validation introduced by CI. According to continuous training (CT), the introduction of new data and a trigger are needed to retrain the model or enhance model performance using online techniques. Additionally, suitable monitoring facilities guarantee that operations are carried out correctly.

III. RESEARCH METHODS

We created the following research questions to help us reach this goal: What are the most recent developments in terms of MLO adoption in practice and the various phases that businesses go through as they develop their MLO practices?

How do case organizations enhance their MLO practices and evolve? To respond to the two RQs, we conducted a systematic literature review [18-20] and the validation case study [22].

A. Systematic Literature Review

Finding, analyzing, and interpreting pertinent research on the subject of interest are the objectives of the SLR [18] [19]. We created search strings in accordance with [18] and browsed five well-known scientific libraries to get answers to the RQs. An overview of the SLR and GLR processes employed in this investigation is shown in Figure 1. For a more thorough examination, we combined and exported pertinent research into an Excel spreadsheet. We incorporated conference and academic articles that detailed MLO in SLR. However, we disregarded studies that were duplicates, published in languages other than English, did not undergo peer review, and were not electronically accessible via the Internet. To offer a thorough description of the current state of practice and practitioner experiences with implementing MLO, we performed the GLR [20]. In contrast to the SLR, the GLR offers the perspective of industry professionals on the subject at hand. Through the use of a “.com” domain name filter, we added studies that address MLO that were published in English and were available in PDF format to the Google Search in GLR. Peer-reviewed scientific publications and other knowledge sources, such as blogs, postings, etc., were eliminated in order to increase the dependability of the GLR’s conclusions. We created an MLO framework and several stages that businesses go through while developing MLO practices based on these research.

B. Validation Case Study

To map organizations to the stages of the maturity model generated from literature studies, we carried out a validation study after [44]. Case study technique is an empirical research strategy built on a thorough examination of a current phenomenon that is challenging to investigate independently in its actual setting [45]. Case studies are used in SE to enhance the SE process and subsequent software solutions by better understanding how and why SE was carried out [46].

The description of each company is provided in Table 1, the particulars (A*, B*, C*, and D* represent self-interview, participant-workshop, joint-meeting, and stand-up meeting respectively) and their roles.

1) Hardware screening: To reduce the amount of hardware returned by customers for repair, the telecommunications firm foresees defects in hardware. They concentrate on sending damaged hardware to the repair facility and returning hardware that isn't defective to the client in this use case.

2) Self-driving Vehicles: The car manufacturer aspires to offer solutions for autonomous mobility. Self-driving cars are the major application to boost productivity. The business must also make sure that these safe-critical use cases have a low failure rate.

3) Defect Detection: The packaging business offers clients packaging solutions in addition to machinery. Defect detection in finished/semi-finished packages is one of the key application cases.
Data Gathering and Analysis: We utilized workshop sessions, meetings, and stand-up meetings at businesses to gather data. They were conducted through video conference in English. All interviews took 45 minutes, meetings and seminars lasted 30 to 60 minutes, and daily stand-up meetings took 15 minutes. We presented the many steps that businesses go through when deploying MLO and verified the MLO framework in example firms. Empirical data was recorded using interview transcripts, workshop notes, meeting notes, and stand-up notes. Later, the primary author distributed them to the other authors for in-depth analysis. To assess and classify the acquired empirical data, we used open coding techniques [47]. Triangulation was utilized to get multiple viewpoints on the subject being studied [48].

IV. THREATS TO VALIDITY

In this investigation, potential validity concerns were taken into account and minimized [49]. Consideration of data from literature review, and the validation enhanced construct validity. The authors and practitioners who contributed to this study are experts in MLO. To gather and validate empirical data, a variety of methods and a variety of sources were employed. By interacting with the other two authors, internal validity concerns brought on by flawed findings as a result of the lead author's bias in data selection or interpretation are lessened. The generalization of the results may be validated and external validity can be reduced by expanding our research to other case firms.

V. LITERATURE REVIEW FINDINGS

ML model creation should include options for parallel experimentation, hyper parameter optimization of the selected model, and evaluation of the model to check that it adheres to the business case. Data scientists may work together on the same code base, which gives them the flexibility to execute the code in various settings and against a range of datasets. This makes scalability, tracking the progress of several studies, and repeatability easier [29].

A model must be integrated with other models and current applications before being deployed to production [30] [41]. The model fulfills requirements while it is in production. When performance declines, keep an eye on the model and turn on the data feedback loop to retrain it. Enable the CI/CD pipeline and continuous retraining through the CT pipeline to perform continuous integration and delivery in a fully developed MLO environment [41] [31].

VI. MLO FRAMEWORK AND MATURITY MODEL

We construct an MLO framework that details the steps necessary for adopting MLO based on the SLR and the GLR. Pictured in Figure 2 is the MLO framework. Three pipelines make up the complete framework: Data pipeline, modelling pipeline, and release pipeline are the first three. Preprocessing of data and feature extraction are carried out following the collection of data from data sources that is pertinent to ML models. The data repository both house versioned data and code. Store the deployable model version in the model registry to keep track of it. Utilizing CI/CD/CT, ML model deployment cycles can be sped up.

MLO Maturity Model: We provide a maturity model that is based on the SLR and the GLR, and it out-lines four stages in which businesses change as they implement MLO practices. The four steps are as follows: a) Automated Data Collection; b) Automated Model Deployment. Fully automated model monitoring is followed by semi-automated model monitoring. These phases document crucial turning points in the implementation of MLO. Each MLO step and the prerequisites needed for a corporation to pass each level are de-scribed below.

A. AUTOMATED DATA COLLECTION

Companies process data, models, deployment, and monitoring manually at this level. As a result of the deployment of MLO, the organization has moved from manual data gathering to automated data collection for (re)training processes. A system is needed for the switch from manual to automated data collecting [22]. Additionally, regardless of variety, volume, or velocity, it necessitates the ability to integrate and process new data sources [21]. Infra-structure resources are also needed for collaborative work and automated data collecting [24, 28]. It is necessary to mimic the feature modification used during training during inference [25]. By tackling data management issues including accountability, transparency, regulation and compliance, and ethics, AI teams may foster trust [27].
B. Automated Data Deployment

At this point, the organizations use a manual methodology for deployment and monitoring. As a result of the implementation of MLO, they move from manual deployment of models and tracking to automated model deployment.

Preconditions: Implementing automated model deployment provisions, particularly across development, quality assurance, and production circumstances [23] [24], can help with the transition. It promotes on-premises, cloud, and edge deployment flexibility [24] [28]. Model hosting, evaluation, and maintenance [22], as well as methods to register, package (containerization [28] [24]), deploy models [23] [20] [29] and integrate [23] are examples of adequate infrastructure options. Experiments [23] [29] [20] and models [21] are being tracked.

![Data Pipeline](image1)

![Modeling Pipeline](image2)

![Package](image3)

![Figure 1 Machine Learning Governance Operations Framework.](image4)

Validating models and data properly [26] helps hasten the adoption of automated models. Automated distribution of retrained models can be facilitated by Canary Deployments [22] and features to save, annotate, discover, and handle models in a single repository [29]. To operationalize and grow AI at this point, multi-talented teams of engineers and ML experts are also required [27].

![Figure 2 Machine Learning Operations Model](image5)

C. Semi-automated Model Monitoring

Companies now use a manual model monitoring system. They can go from manual model monitoring to partially automated model monitoring with the aid of MLO.

Preconditions: To make this transformation, there must be facilities for triggering [30] when performance declines and tools for diagnosing, tracking, and resolving model drift [23] [27] [26]. Additionally, management and monitoring of models based on drift [28] and the capacity for continuous model tracking [21] require automation scripts. MLO experts must have access to visual tools [34], as well as specialized and central dashboards [28] [27] [28] for simple model monitoring. To ensure data changes, it also needs data orchestration pipelines, rule-based data governance, feedback loops, and constant model retraining [30]. Additionally, a system that automatically trains models in production with new data based on active pipeline triggers and feedback loops should exist [28].

D. Fully Automated Model Monitoring

The firms have models deployed and being monitored, and when performance declines, an alarm is generated. They make the shift to fully automated model monitoring by utilizing MLO.

Preconditions: The organization needs CI/CD integration with automation and orchestration for this
transformation [24] as well as a CT pipeline to retrain models when performance declines [21]. Model certification [22] [23], governance and security controls [23] [24] [26], model explainability [23] [26], auditing of model usage [24] [23], and repeatable workflow and models [26] are all necessary for this transformation. End-to-end quality assurance tests and performance audits need to be possible [29]. Data pipelines should provide confidence that data security and privacy standards are implemented [30], and production models should be trained on newer data using the same data, methods, and code that were used to construct the original [30].

VII. Model Validation

Using the following three case studies, the proposed model is validated.

A. Company # A

The business scheduled an introductory meeting with team members to go through reasonable expectations for MLO prior to adopting it. P2 asserts that although practitioners initially need to spend a lot of effort developing the architecture, communicating, and debating MLO, the final result is a considerable decrease in human labor and end-to-end automation. The main objectives are to accomplish Automation, Versioning of datasets, models, Traceability and Reproducibility.

P1: “Although we have partially implemented the idea of MLO, we have already started. We are attempting to determine what is in place, what is not, and how to put things in place that are not currently in place.”

Practitioners must have a pipeline for data in place and training data registration capabilities when working with data. This procedure was manual in the business prior to the implementation of MLO. Data schema has to be checked in order to guarantee the integrity of the data pipeline. To make it easier for users to compare various models and visualizations, the organization is considering implementing DVC (Data Version Control). When working with models, practitioners maintain the model’s quality while monitoring model performance by adjusting hyper parameters.

Experts in case companies Put more of a focus on knowing how a project functions, particularly when it comes to the idea of model deployment. They may then test the project and the data using a smaller data set (for example, 10% of the total size of the data set) to ensure everything is working as it should. Practitioners will transfer the models to the production environment if they perform as expected in the development environment. They must take into account other models that are already in use in order to merge these models. Because of this, practitioners must spend some time learning how to input prior data into the system. The business utilizes Graff Ana for model monitoring and Tableau for data visualization.

In the instance of firm, A, the model pipeline is not entirely automated, and the data pipeline is still fairly young. The model serving pipeline, on the other hand, is quite compatible with MLO. To compare models, the business uses dataset versioning. They formerly used Gate for versioning before switching to Art factory.

W5: "We currently have data and model versioning, but we may eventually move to versioning the entire pipeline, configurations, etc."

P1: “There are some continuous tasks we perform as part of our daily work, and some of these tasks can be automated. Instead of developing test cases that can be highly automated, a data scientist should concentrate on feature engineering, model training, etc.”

B. Company # B

Case firm B is likewise attempting to use MLO to their situation. The business gathers data from actual automobiles or creates it using simulations when dealing with data. They add information to the logs after entering this data into a logging system. They check the picture quality after labelling. Through an API, practitioners can access the logged data, which is transferred from hard drives to servers. The organization is excited to guarantee annotation uniformity throughout the project. The dataset is pre-processed and divided into training and validation once it has been annotated.

They train neural networks that spin functional nodes, and reallocate them after training when it comes to models. They support the experiments, verify the mod-els, and speed up inference by using model pruning. The practitioners deploy the model to art factory after converting it with ONXX. The model may be applied to vehicles once it has been launched in Art factory. They essentially use the CI/CD loop to deploy the pipeline. Additionally, they modify the validation set to reflect new data, domains, etc. For scalability, the business is considering switching from on-site to cloud services. The business produces artefacts.

W8:” In order to perform inferences, we require a tool chain that includes data selection, development, and deployment on the target devices. For some artefacts, we rely on different groups inside the organization. Take the logging system, for instance. The group also constructed the remaining artefacts.”

C. Company # C

The goal of Case Company C is to achieve completely automated MLO. The company has a controllable deployment procedure, standardized and high-quality model development, and a model lifecycle that is now in use. Achieving a) Scale with excellent availability, b) Flexibility and extensibility, c) Integration, d) Automation, and e) Maintainability are the fundamental architectural concepts of Case Company C.
When working with data, the company either uses cam-eras to take pictures of packages or simulations to create them. Before being used for training, the collected data is kept in a data lake. Before choosing the most appropriate model and implementing hyper-parameter tuning, the practitioners test out several algorithms. They utilize GPUs for training to reduce needed time. The business integrates DevOps practices into its ML systems. Docker containers are used to bundle and deliver the models to production. They use Kubernetes to automate scalability and deployment. The business has systems in place for monitoring data, models, and experimentation. They have a modelling system in place, and models may be deployed on-premises, at the edge, or in the cloud. Dashboards may be used to visualize the model monitoring, and as performance declines, models can be re-trained. The business also employs a tool chain for model creation and deployment.

VIII. VALIDATION STUDY

This study emphasizes the growing interest in MLO and the expanding use of these techniques in systems that depend heavily on software. More pertinent GLR re-search [20, 21] on MLO are retrieved from the literature than SLR studies. The case firm A is shown to be in between the steps of automated data gathering and automated model deployment in Figure 4. This is due to the fact that, despite this company's highly developed modelling and deployment pipelines, its data pipeline is still in its infancy. The business also plans to version the release pipelines, models, and data. The difficulties Case Company A had in the early stages of MLO are the same difficulties we noted in literature assessments. The difficulties with the data flow that firm A is facing are typical and have been highlighted by other businesses during investigations that were a part of the GLR. business B is positioned at stage one of the maturity model, Automated Data Collection, much like case business A was.

This is due they want to make sure that annotations are consistent throughout the project. On the other hand, they contain features for gathering data from many sources, running important data queries and algorithms, tracking experiments, etc. They are at stage one since their data flow is not entirely automated.

The maturity model places Case Company C, at stage two, Automated Model deployment. They use Docker containers for deployment, data lakes for data collection, frameworks for executing experiments, and DevOps approaches in their application. They also offer systems for deploying models to the edge or cloud. When a model degrades, model management and model re-training are started. They are at stage two even though they have automated model deployment tools because they want to accomplish model governance, fairness, generalizability, and explain ability.

IX. CONCLUSION

To enable continuous development, deployment, and delivery of ML systems, businesses apply DevOps techniques. In this study, we build a framework that distinguishes the MLO adoption activities and the stages that enterprises go through as they advance. We test this paradigm in three firms that create embedded devices with a high concentration of software, and we show how these companies have adopted and incorporated MLO into their large-scale software development organizations. In order to broaden our study and validate our findings, we want to include more case companies and experts in subsequent studies.

This is encouraging since it shows that more businesses are working to implement completely automated MLO. We observe that the feature store, data archive, code archive, metadata archive, model database, and responses loops can hasten the transition of models from prototyping to production stage based on findings from the literature. They thereby support automation, versioning, explain ability, and traceability.

In order to enhance its validity for future, the target is to explore different industry sectors and refine the framework based on additional empirical evidence.

REFERENCES


