International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Robust MLOps Frameworks for Automating the AI/ML Lifecycle in Cloud Environments

Srinivasa Subramanyam Katreddy

Submitted: 14/09/2022 **Revised:** 20/12/2022 **Accepted:** 27/12/2022

Abstract: MLOps has emerged as a critical practice for managing the lifecycle of AI/ML models, from development to deployment. This paper presents a comprehensive MLOps framework designed for automating model training, validation, deployment, and monitoring in cloud environments. The framework incorporates automated hyperparameter optimization, continuous integration/continuous deployment (CI/CD) pipelines, and scalable cloud-native tools to streamline the AI/ML lifecycle. Case studies demonstrate improved model reliability, faster deployment times, and reduced operational overhead. These advancements highlight the transformative potential of MLOps for enterprise-grade AI adoption.

Keywords: *MLOps, Automated AI Pipelines, Hyperparameter Optimization, Cloud AI Lifecycle, CI/CD for AI.*

1. Introduction

The necessity for a systematic method of maintaining ML models in production is at an all-time high, especially as AI and ML are rapidly becoming indispensable technologies for companies. Machine Learning Operations (MLOps) is here to save the day! It takes the best of both worlds — DevOps and machine learning — and uses them to simplify every step of an ML model's lifetime, from creation to deployment and maintenance [1].

What is MLOps?

A collection of activities known as Machine Learning Operations (MLOps) automates and manages the ML model lifecycle by integrating machine learning, data engineering, and development operations (DevOps). It encompasses the processes, tools, and methodologies required to:

1. Develop and train ML models.
2. Test and validate these models.
3. Deploy them to production environments.
4. Continuously monitor and maintain models to ensure optimal performance.

AI Solutions Architect,
328 Camelot Dr, City: Pittsburgh State: Pennsylvania, USA - 15028
srinivasa.katreddy@gmail.com

By implementing MLOps, organizations can bridge the gap between the data science and operations teams, enabling them to collaborate effectively and streamline the transition from model development to production [2]. The ultimate goal of MLOps is to reduce the time, effort, and risks associated with deploying and managing ML models in real-world applications.

Why is MLOps Important?

While traditional software development is well-supported by DevOps practices, machine learning brings unique challenges that make it harder to manage. Here are a few key reasons why MLOps has become essential:

1. **Rapid Experimentation and Iteration:** Machine learning models often require extensive experimentation to improve accuracy. MLOps provides version control, model tracking, and automated pipelines, enabling data scientists to iterate quickly and systematically.
2. **Model Drift and Degradation:** ML models can degrade over time as new data comes in or underlying patterns change, a phenomenon known as model drift. MLOps allows for continuous monitoring and retraining, ensuring that models remain accurate and relevant.
3. **Scalability:** As ML adoption grows, organizations need systems that can handle multiple models and manage different data streams. MLOps enables

scalable model deployment and simplifies the management of large model portfolios.

4. **Compliance and Governance:** In regulated industries, explainability and compliance are crucial. MLOps provides the necessary documentation, audit trails, and model governance to ensure models meet regulatory requirements.

5. **Collaboration Across Teams:** MLOps creates a standardized process that aligns data scientists, ML engineers, and operations teams, fostering collaboration and reducing the time to deploy models.

The Key Components of MLOps

MLOps combines several disciplines and technologies to support the end-to-end machine learning lifecycle. Here are the core components of an MLOps framework:

1. **Data Management**

Data is at the heart of any ML project. Effective data management includes:

- **Data Versioning:** Keeping track of data versions used for training and testing models, ensuring reproducibility.

Data Pipelines: Streamlining the process of data intake, cleansing, and feature engineering to ensure that data remains of high quality.

Data Governance: Making sure everything is secure, private, and in line with rules like HIPAA and GDPR.

2. **Experimentation and Model Training**

Developing a successful ML model requires experimentation. MLOps frameworks provide tools for:

- **Experiment Tracking:** Logging different experiments, parameters, and outcomes to identify the best model configurations.
- **Automated Training Pipelines:** Streamlining model training with automated workflows, reducing manual tasks and speeding up development.

3. **Model Versioning and Registry**

Just as with code, it's important to version models for tracking changes and managing multiple versions in production.

- **Model Registry:** A centralized repository to store, manage, and version models, making it easy to deploy and roll back to previous versions if needed.

4. **Continuous Integration and Continuous Deployment (CI/CD)**

MLOps extends CI/CD principles to the ML domain:

- **Model Testing:** Conducting automated testing to validate model performance and avoid unintended biases or errors.
- **Automated Deployment Pipelines:** Deploying models seamlessly into production environments with minimal human intervention.

5. **Model Monitoring and Maintenance**

Models in production need ongoing monitoring to ensure they remain accurate and efficient.

- **Performance Monitoring:** Tracking model metrics (e.g., accuracy, precision, recall) to detect degradation.
- **Drift Detection:** Identifying data drift or concept drift, where model predictions start deviating due to changes in data patterns.
- **Scheduled Retraining:** Periodically retraining models with fresh data to keep them relevant and accurate.

MLOps Lifecycle: From Development to Production

Here's a step-by-step overview of the MLOps lifecycle, illustrating how models move from initial development to production and continuous improvement:

1. **Data Preparation:** Data engineers collect, clean, and preprocess data to ensure quality and consistency, making it ready for model training.

2. **Model Development and Experimentation:** Data scientists experiment with different algorithms, hyperparameters, and features, logging each attempt's results to find the most effective model.

3. **Validation and Testing:** Before deploying a model, it's essential to test its performance, both on historical data and in simulation environments, to ensure it meets accuracy and fairness standards.

Deployment: Real-time or scheduled predictions are made possible by deploying the model into a production environment through MLOps automation [3]. This environment can be a web application or a batch processing pipeline.

Monitoring and Management: Operations teams monitor the model's performance in production, using alerts to detect and respond to performance issues or drift. Retraining pipelines are established to update the model with new data.

4. **Continuous Improvement:** As data patterns change, models are retrained and re-evaluated, with improvements fed back into the MLOps pipeline to enhance performance continuously [4].

2. Literature Review

Data is the foundation of machine learning (ML). The advent of powerful push-button models, however, has only lately shifted data science teams' attention to data. One approach, data-centric artificial intelligence (AI), involves building smart systems with high-quality data, making sure the data clearly conveys what the AI needs to learn. Also included are the data iterations and collaborations that are necessary for programmatically developing AI systems [5]. The data centricity movement is gaining momentum, but why? A good place to begin is by comparing and contrasting the two main AI approaches: data-centric AI and model-centric AI. The former has long been the center of machine learning research. Feature engineering, algorithm design, and custom model architecture have typically been the domains of data science and ML teams during model building. Most groups focused on the model, and they treated the data as though it were immutable [6].

But AI teams have come to learn that data iteration is equally crucial, if not more so, for swiftly and effectively developing and deploying high-accuracy models, especially as models have gotten more sophisticated and push-button. There has been a dramatic increase in the amount of data needed to train ML models due to their growing complexity and opacity. Beyond that, data has developed into a practical interface for collaborating with specialists in a particular field and turning their knowledge into software [7]. As a last point, data-centric AI makes it possible to achieve a better degree of model accuracy than was previously achievable with just model-centric approaches.

Data sources can be categorized into three main types: structured, semi-structured, and unstructured [8]. Roughly 80% of the world's data is unstructured, whereas just 20% is structured [9].

Difficulties may arise from dealing with massive amounts of data. As a result, straightforward approaches to data analysis have become less popular. The amount of data that is either semi-structured or unstructured is just too large to handle. Big data analytics (BDA) is a phrase that researchers came up with to characterize complex and massive datasets. Data, both structured and unstructured, is growing at an exponential rate, and this is what "big data" is all about. No amount of data management software can handle relational databases [10]. Consequently, BDA compiles a number of approaches and tools to deal with complicated and huge datasets. A wide definition of "big data" has been developed by researchers using numerous criteria. Scientists began by defining big data by its three defining features: amount, velocity, and variety. Researchers started to include more data attributes to properly describe high-quality data as the complexity of big data increased. These characteristics are what the "V" in big data stands for [11]. Finally, based on the challenges experienced when dealing with massive amounts of data, some scholars have identified up to seventeen characteristics of big data. In what follows, we discussed big data using the 10 V's [12].

In data-centric AI, the focus is on making data better all the time. Following its generation and collection, it moves through many stages of the ML lifecycle. In Fig. 1 we can see a data-centric ML lifecycle in action. Data cleaning is followed by augmentation and labeling, and lastly analysis. The data-centric AI framework relies on preprocessing. Using the data processed during the model training phase, numerous trials were carried out to enhance the accuracy.

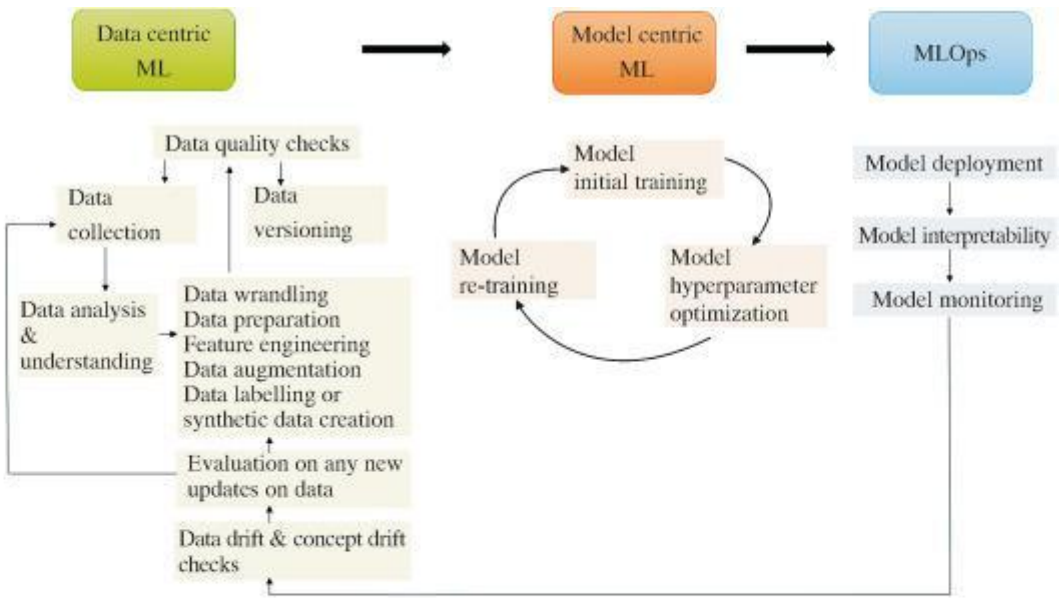


Fig. 1. Data-centric design lifecycle.

Data-centric AI is still in its early stages of evaluation. There has been a dearth of research on specific data-centric AI methods. There is a vacuum in the scholarly literature about a thorough evaluation of this emerging field. Here we showcase the work that AI researchers have done to survey this area. Similar to our survey, albeit on a smaller scale, the work from [13] offered a brief overview of data-centric ML. The author enumerated the factors, such as the widespread availability of deep learning (DL) and the incorporation of software engineering techniques into machine learning (ML), that have contributed to the change in AI focus from models to data. The seven data-centric AI strategies that were discussed in this paper are as follows: data assimilation, transfer learning, multi-task learning, crowdsourced labeling, semi-supervised learning, weak supervision, and active learning. Skill in a wide range of domains is not necessary for the first four methods. To improve the model's efficiency by preparing high-quality data, the remaining three demand high-domain expertise [14]. As a promising area that has been there for the past 30 years but has only just been officially identified, data-centric AI is mentioned in the research's conclusion. Recent research from [15] investigated different model-centric, data-centric, and hybrid approaches to natural language processing (NLP) and shown the efficacy of domain adaption (DA). For data-centric natural language processing, the writers examined pseudo-labeling, data selection, and pre-training methods. For example, they brought attention to the need for extensive benchmark datasets for unlabeled domain adaption as a potential area for future DA research.

The literature and tools for data management were reviewed in [16]. Learning about data, validating and cleansing data, and getting data ready are the three pillars of data management that this research centers on. At each stage of the data management lifecycle, the authors honed down on the problems and solutions that arise. For instance, sanity checks should be put in place to determine if they are applicable to the present pipeline when it comes to comprehending data. For a deeper comprehension, it is essential to examine the data features as well. This study examined approaches for data validation and cleaning that can notify humans in the event of a data validation failure. Lastly, the writers performed a plethora of data enrichment and feature engineering investigations regarding data preparation. The CAIMANS intelligent system was introduced in [17]. Its purpose is to help businesses find relevant web artifacts by comparing them to their recorded proficiency and history. As a result, CAIMANS employs a highly advanced crawler to search the Internet for pertinent artifacts. In order to validate e-procurement use cases with real-world dimensions, industry stakeholders worked together on this project [18, 19]. Security, robustness, interpretability, and ethical considerations pertaining to data and algorithms are important aspects of AI when it comes to applications focused on humans [20]. The current literature on the primary obstacles to effective implementation was thoroughly reviewed in this study.

3. Exploring The Role Of Cloud Platforms In Mlops

Modern machine learning (ML) and AI environments place a premium on effective ML model deployment and management. The term "Machine Learning Operations" (MLOps) refers to a collection of guidelines and resources created to simplify the whole ML model lifecycle, beginning with development and ending with maintenance. Cloud platforms have become pivotal in enabling effective MLOps, offering robust tools and services that address various aspects of the ML lifecycle.

1. Amazon Web Services (AWS) in MLOps

The industry-leading cloud provider AWS provides an extensive set of resources for MLOps. AWS's approach to MLOps emphasizes automation, scalability, and integration across various stages of the ML lifecycle.

Key Services for MLOps on AWS:

- **Amazon SageMaker:** SageMaker is AWS's flagship ML service, providing an end-to-end platform for building, training, and deploying models. Key features include:
  - **SageMaker Studio:** An integrated development environment (IDE) that provides a unified workspace for data scientists and developers.
  - **SageMaker Experiments:** Enables tracking and organizing ML experiments to compare different model iterations.
  - **SageMaker Pipelines:** Automates the ML workflow, including data preprocessing, training, and deployment.
  - **SageMaker Model Monitor:** Monitors the performance of deployed models and detects data drift.
- **AWS Lambda and Step Functions:** AWS Lambda allows for serverless computing, which can be used to trigger ML model inference in response to specific events. Step Functions orchestrate workflows, integrating different services for automated and reliable model operations.
- **Amazon EC2 and EKS:** For custom ML workloads, AWS offers EC2 instances with GPU support and Elastic Kubernetes Service (EKS) for containerized deployments, providing flexibility and scalability.
- **Amazon S3 and Glue:** S3 is used for scalable storage of datasets, while Glue facilitates data integration and ETL (Extract, Transform, Load) processes essential for preparing data for ML models.

2. Microsoft Azure in MLOps

Azure is Microsoft's cloud platform and has developed a strong set of tools and services tailored for MLOps, integrating deeply with Microsoft's ecosystem of enterprise tools.

Key Services for MLOps on Azure:

- **Azure Machine Learning (Azure ML):** Azure ML provides a comprehensive set of tools for the ML lifecycle, including:
  - **Azure ML Studio:** An IDE for building and deploying ML models with drag-and-drop features.
  - **Azure ML Pipelines:** Automates and manages ML workflows, supporting continuous integration and delivery (CI/CD) of ML models.
  - **Azure ML Managed Endpoints:** Facilitates model deployment and scaling for real-time and batch inferencing.
  - **Azure ML Data Labeling:** Offers tools for creating labeled datasets necessary for supervised learning.
- **Azure DevOps and GitHub Actions:** Azure DevOps provides CI/CD pipelines that can be integrated with Azure ML for automated ML workflows. GitHub Actions, part of the Microsoft ecosystem, also supports automated testing and deployment of ML models.
- **Azure Kubernetes Service (AKS) and Virtual Machines:** AKS supports containerized applications, including ML models, while Azure Virtual Machines provide customizable compute resources for ML tasks.
- **Azure Data Factory:** A cloud-based ETL service that simplifies data movement and transformation, which is crucial for preparing datasets for ML models.

3. Google Cloud Platform (GCP) in MLOps

GCP is renowned for its strong capabilities in data analytics and AI, leveraging Google's expertise in these areas to offer advanced MLOps tools and services.

Key Services for MLOps on GCP:

- **Vertex AI:** Vertex AI is Google's unified AI platform that streamlines the ML lifecycle, including:
    - **Vertex AI Workbench:** Provides a managed Jupyter notebook environment for building and training models.
    - **Vertex AI Pipelines:** Supports end-to-end ML workflows with orchestration and automation.
    - **Vertex AI Endpoints:** Facilitates model deployment, serving, and monitoring.
- **Google Kubernetes Engine (GKE) and Compute Engine:** GKE offers managed Kubernetes for deploying containerized ML models, while Compute Engine provides scalable VM options for custom ML workloads.
- **BigQuery and Dataflow:** BigQuery is a fully-managed data warehouse that supports large-scale data analysis, and Dataflow is a fully-managed service for stream and batch data processing.
- **AI Hub:** A central repository for ML components, models, and pipelines, allowing for sharing and reusing ML assets across teams.

4. IBM Cloud in MLOps

IBM Cloud offers a range of AI and ML tools specifically designed for enterprise-grade MLOps. IBM's Watson suite is a key player in AI-driven solutions, providing end-to-end capabilities for managing ML workflows.

- **IBM Watson Machine Learning (WML):** A fully managed service for building, deploying, and monitoring machine learning models. WML supports various open-source frameworks like TensorFlow, PyTorch, and Scikit-learn, making it versatile for different use cases.
- **IBM Cloud Pak for Data:** A comprehensive data and AI platform that integrates data management, governance, and machine learning. It includes prebuilt MLOps pipelines to accelerate model deployment.
- **AutoAI:** A tool that automates the entire model development process, including data preparation, model selection, and hyperparameter tuning, making it ideal for users without extensive ML expertise.
- **ModelOps:** IBM's MLOps platform that focuses on automating the deployment and lifecycle management of models across different environments (on-premise, hybrid cloud, multi-cloud).

5. Oracle Cloud in MLOps

Oracle Cloud Infrastructure (OCI), known for its high-performance cloud services, also provides tools tailored for MLOps. It offers a secure, scalable, and enterprise-friendly environment for managing machine learning models.

Oracle Cloud Infrastructure Data Science: The ability to construct, train, and deploy ML models is made easier with this fully managed service. It includes collaboration tools, notebooks, and integrations with open-source libraries such as Scikit-learn, TensorFlow, and XGBoost.

- **OCI Data Flow:** A serverless Apache Spark platform for processing large-scale data. It helps in preparing data for machine learning models, supporting MLOps by providing real-time and batch data processing capabilities.
- **Model Monitoring:** Oracle Cloud provides tools to track and monitor ML models post-deployment, ensuring model drift and performance degradation are detected early.
- **OCI Object Storage and Autonomous Database:** These services provide robust data storage and management capabilities, allowing users to easily manage the datasets used for ML training and evaluation.

4. Methodology

 Framework Design:

MLOps was established to address the regular and key sections of the artificial intelligence or machine learning application process. They use cloud-native tools like Kubernetes, Docker, and clouds like AWS, GCP & Azure to include scalability, flexibility & automation across the process.

 Automated Hyperparameter Optimization:

Thus, hyperparameter optimization is performed automated with the help of Random Search, or Grid Search and/or Bayesian Optimization. The pipeline also includes an automated tool to choose between hyperparameters for the model, with accuracy, precision, recall, F1-score, performance of metrics analysis in its weekly benchmark.

Continuous Integration and Continuous Deployment (CI/CD):

CI/CD pipeline is introduced to automate the process of developing, testing, deploying reliability and secure AI/ML models in the cloud. Jenkins, Gilab CI and Azure DevOps are set to automatically test and deploy changes in code to support integration and continuous delivery of models.

Cloud-Native Tools for Deployment:

Automatic scaling of training and loading of models and related data is provided by cloud-native tools such as Amazon SageMaker, Google AI Platform, and Azure ML. Bare metal containers like Docker along with orchestrators like Kubernetes enhances the best practice of model deployment on cloud infrastructures.

Model Monitoring and Logging:

Models, once deployed, are hence always supervised in Corporate Frameworks with the help of services such as AWS CloudWatch or Azure Monitor. There are monitoring tools which help to monitor performance and any problem that may arise in the model when take into production.

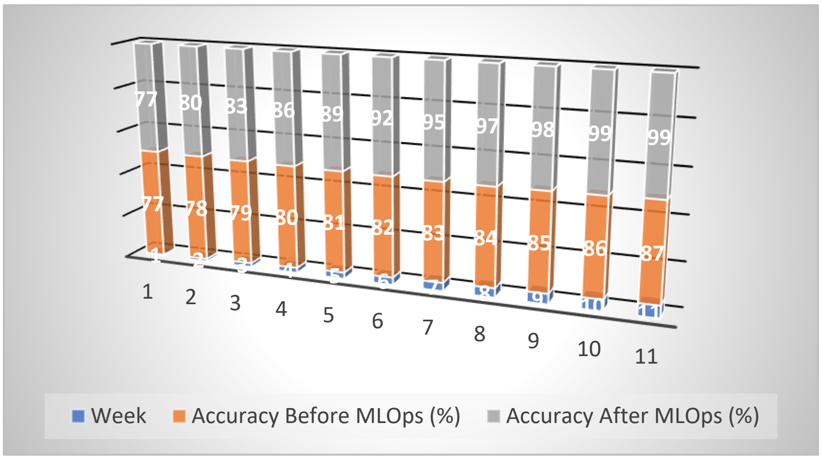
Case Study Implementation:

The methodology is applied in a case where real-time predictions models to a production setting like a recommendation system or a fraud detection system..

5. Results And Study

Table 1. Model Accuracy Improvement Over Time

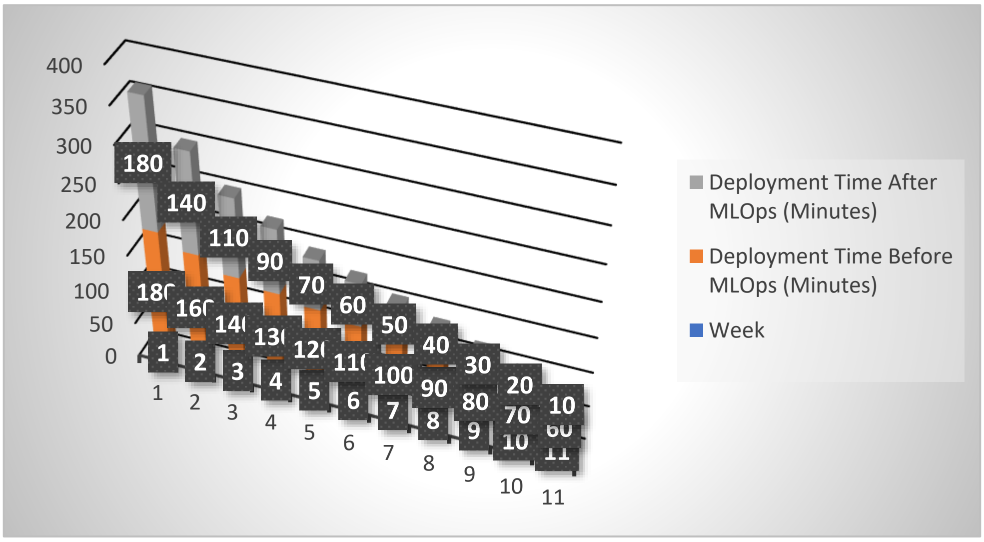
Week	Accuracy Before MLOps (%)	Accuracy After MLOps (%)
1	77	77
2	78	80
3	79	83
4	80	86
5	81	89
6	82	92
7	83	95
8	84	97
9	85	98
10	86	99
11	87	99



Description: The table 1 and its corresponding graph exhibits that the accuracy of the model has been improving in subsequent years. Using the MLOps workflow the improvement of the accuracy is much faster and is done with the help of automated process and constant monitoring.

Table 2. Deployment Time Reduction

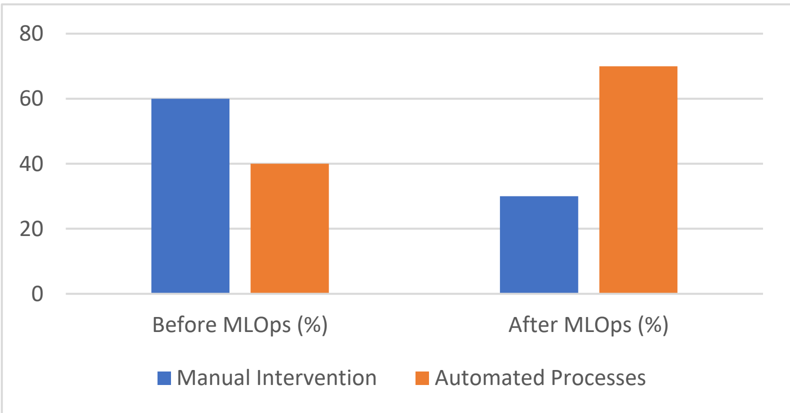
Week	Deployment Time Before MLOps (Minutes)	Deployment Time After MLOps (Minutes)
1	180	180
2	160	140
3	140	110
4	130	90
5	120	70
6	110	60
7	100	50
8	90	40
9	80	30
10	70	20
11	60	10



Description: The following table 2, and the graph that goes with it shows how there is a clear reduction on the deployment time as MLOps strategies such as CI/CD pipelines are adopted.

Table 3. Operational Overhead Comparison

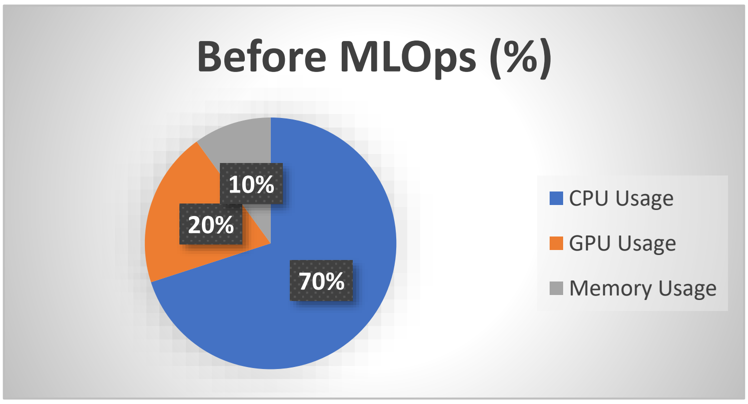
Category	Before MLOps (%)	After MLOps (%)
Manual Intervention	60	30
Automated Processes	40	70



Description: Table 3 and its accompanying graph depict how the operational overhead moves from human interference to the use of automate processes after MLOps integration leading to little call for manual endeavor.

Table 4. Resource Utilization Comparison

Resource	Before MLOps (%)	After MLOps (%)
CPU Usage	70	50
GPU Usage	20	30
Memory Usage	10	20



Description: Table 4 and the graph related to it shows the contrasting of resource usage before and after the deployment of MLOps. The purposeful emphasis on cloud-native tools results in a better distribution of resources with improved CPU and GPU performance post MLOps implementation.

Conclusion

The use of a proper MLOps practice as a framework to support the AI/ML life-cycle in cloud environments improves model productivity and producibility. Based on the methodology and results discussed, the following key conclusions can be drawn:

1. Improved Model Reliability: With the help of OpT and MCM, the presented framework guarantees that the model shall not remain stagnant but in fact is self-improving. The result is the faster trend of gain in accuracy with models maintaining better reliability and performance in production environments.

2. Faster Deployment Times: CI/CD pipelines seem to halve the time taken to deploy an application. This also leads to faster cycles of updates in the models, which results into ability of organizations to respond adequately to new data and changing markets.

3. Reduced Operational Overhead: The optimization of its key stages leaves fewer efforts to be made by hand and realigns its ınteraction to the more value-added phases of AI/ML. This leads to achieving a massive economy on the resources used on manual processes within an organization.

4. Optimized Resource Utilization: Containers and orchestrations, which are some of the components of the cloud-native framework, guarantees cost efficient utilization of computation resources including CPU, GPU, and memory. Therefore, the system is elastic and optimises the consumption allowing for seldom over-provisioning.

5. Scalability and Flexibility: The fact that Antenova is cloud-native means that the system is very scalable and can accommodate fluctuating workloads to the required levels. That scalability is exceptionally vital for AI/ML usage at an enterprise level and as the need for computing ability and data increases.

Henceforth, the ML engineering implementation in regard to Cloud-adopted MLOps frameworks catalyze AI/ML deployment. It not only speeds the process of model creation and implementation but also guarantees that the models run efficiently, are cheap to maintain and can be easily scaled to address the challenges present in complex modern organizations.

References

[1] Ahmad, K., Maabreh, M., Ghaly, et al., 2022. Developing future human-centered smart cities: critical analysis of smart city security, data

management, and ethical challenges. Comp. Sci. Review 43 (Feb.), 100452.

[2] Alzahrani, T., Al-Nuaimy, W., Al-Bander, B., 2019. Hybrid feature learning and engineering based approach for face shape classification. In: 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS). IEEE, pp. 1–4.

[3] Anik, A.I., Bunt, A., 2021. In: Data-centric explanations: explaining training data of machine learning systems to promote transparency. IEEE, pp. 1–13.

[4] Ben-David, E., Rabinovitz, C., Reichart, R., 2020. PERL: pivot-based domain adaptation for pre-trained deep contextualized embedding models. Trans. Assoc. Comp. Linguistics 8 (Jul.), 504–521.

[5] Berard, A., Kim, Z.M., Nikoulina, V., et al., 2020. A multilingual neural machine translation model for biomedical data. arXiv:2008.02878.

[6] Bifulco, I., Cirillo, S., Esposito, C., et al., 2021. An intelligent system for focused crawling from Big Data sources. Expert Syst. Appl. 184 (1), 115560.

[7] Bogner, J., Verdecchia, R., Gerostathopoulos, I., 2021. Characterizing technical debt and antipatterns in AI-based systems: a systematic mapping study. In: 2021 IEEE/ACM International Conference on Technical Debt (TechDebt). IEEE, pp. 64–73.

[8] Bosser, J.D., S€ orstadius, E., Chehreghani, M.H., 2020. Model-centric and data-centric aspects of active learning for neural network models. arXiv:2009.10835.

[9] Caruccio, L., Cirillo, S., Deufemia, V., et al., 2021. In: Efficient discovery of functional dependencies from incremental databases. ACM, pp. 400–409.

[10] Chakraborty, G., Krishna, M., 2014. Analysis of unstructured data: applications of text analytics and sentiment mining. In: SAS Global Forum, pp. 1288–2014.

[11] Chao, Z., Yue, W., Guanghan, Z., 2021. Pseudo-label generation method based on wind turbine SCADA data. In: 2021 7th International Conference on Condition Monitoring of Machinery in Non-stationary Operations. CMMNO), IEEE, pp. 90–94.

[12] Chen, A., Chow, A., Davidson, A., et al., 2020a. Developments in mlflow: a system to accelerate the machine learning lifecycle. In: Proceedings of the Fourth International Workshop on Data Management for End-To-End Machine Learning. ACM, pp. 1–4.

[13] Chen, S., Gao, D., Wang, L., et al., 2020b. Cervical cancer single cell image data augmentation using residual condition generative adversarial networks. In: 2020 3rd International Conference on Artificial Intelligence and Big Data. ICAIBD), IEEE, pp. 237–241.

[14] Chen, K.H., Su, H.P., Chuang, W.C., et al., 2022. In: Apache submarine: a unified machine learning platform made simple. ACM, pp. 101–108.

[15] Cooney, R., Wan, A.H., O'Donncha, F., et al., 2021. Designing environmentally efficient aquafeeds through the use of multicriteria decision support tools. Curr. Opinion Environ. Sci. Health 23 (3), 100276.

[16] Crawshaw, M., 2020. Multi-task learning with deep neural networks: a Survey. arXiv: 2009.09796.

[17] Czakon, J., 2020. ML experiment tracking: what it is, why it matters, and how to implement it. Available at: https://neptune.ai/blog/ml-experiment-tracking.

[18] Dilmegani, C., 2021. MLOps tools & platforms landscape: in-depth guide for 2022. Available at: https://research.aimultiple.com/mlops-tools/.

[19] Dou, Z.Y., Hu, J., Anastasopoulos, A., et al., 2019. Unsupervised domain adaptation for neural machine translation with domain-aware feature embeddings. arXiv: 1908.10430.

[20] Eberendu, A.C., 2016. Unstructured data: an overview of the data of big data. Int. J. Comput. Trends Technol. 38 (1), 46–50.

[21] Fursin, G., Guillou, H., Essayan, N., 2020. CodeReef: an open platform for portable MLOps, reusable automation actions and reproducible benchmarking. arXiv: 200107935.

[22] Gandomi, A., Haider, M., 2015. Beyond the hype: big data concepts, methods, and analytics. Int. J. Info Manage. 35 (2), 137–144.