Machine Learning Operations (MLOps) for Generative AI

Atchaya N
Bharathiar University

Praveena R
Bharathiar University

Sangeetha K
Bharathiar University

Sashvatha S
Bharathiar University

Sivaranjani S
Bharathiar University

Abstract:

MLOps (Machine Learning Operations) represents the intersection of machine learning and DevOps practices, aimed at automating the deployment, monitoring, and management of machine learning models in production environments. This paper explores the various levels of MLOps automation, compares the methodologies of custom ML and AutoML, and presents a case study on the implementation of MLOps for fraud detection. We also discuss the advantages, limitations, and real-world applications of these methodologies in various domains. Through this analysis, we aim to provide insights into optimizing machine learning workflows and enhancing model deployment processes.

Keywords:

MLOps, machine learning, DevOps, automation, custom ML, AutoML, continuous integration, continuous delivery, fraud detection, data pipelines, model deployment, machine learning workflows.

I. INTRODUCTION

In the rapidly expanding domain of machine learning, the deployment and management of models in production environments have become critical challenges. Traditional approaches to deploying machine learning models often lack scalability and reliability, leading to increased operational complexity and longer development cycles. Machine Learning Operations (MLOps) has emerged as a vital discipline that integrates machine learning practices with DevOps methodologies to automate and streamline the deployment, monitoring, and management of models in production.

The task of deploying AI models to production requires processes after the business model has been determined to ascertain the success rate. These processes provide ML models for production and can be accomplished manually or with the aid of automated workflows. As a result, three different levels of MLOps are defined namely, MLOps Level 0, MLOps Level 1, MLOps Level 2

A. MLOps Level 0

The fundamental level of maturity, or MLOps level 0, refers to simple workflows that are manually script-driven at every stage of the machine learning lifecycle. This level of MLOps suffers from sparse release iterations because it expects the generated model to not change very often. The concepts of CI/CD are not needed as there is no automation, resulting in lack of active performance monitoring. When models are repeatedly changed or trained, the manual-driven process may persist, but in practice, models are regularly changed or trained, demanding regular iterations.

B. MLOps Level 1

The steps of machine learning experiments are orchestrated at MLOps level 1, to automate the ML pipeline solely to undertake continuous model training. This enables it to provide model prediction services continually and automates retraining models in production using new data. As a result, model deployment setups and continuous model delivery are automated, allowing to leverage trained and validated models as a prediction service for making the online predictions. However, in order to test new ideas and quickly release new implementations of ML components, the need for CI/CD solutions to automate the creation, testing and deployment of ML pipelines is critical.

C. MLOps Level 2

Level 2 of MLOps includes automation of a CI/CD pipeline for more rapid and reliable updating of pipelines in production,which requires a more robust automated system. where staged experiment phases can be carried out iteratively for trainingnew algorithms and models. It results in the output of pipeline steps' source code, subsequently pushed into the source repository. Next step involves continuous integration (CI), where the source code is built and undergoes various run tests, resulting in package executables and artifacts being deployed later. Finally, in continuous delivery (CD), the artifacts created in CI phases are deployed to the target environment, resulting in the implementation of updated AI models through the generated pipeline. This pipeline is automated in production and runs according to a schedule or a trigger. The generated trained models are then uploaded to the model registry, and this stage is also known as automatic triggering. It also includes CD, which serves prediction as a service. So, once a model prediction service has been installed, the statistics on model performance based on real-time data are collected. The result of this stage is a trigger to execute pipelines and start a new trial cycle.

This paper explores the application of MLOps for two distinct approaches: custom machine learning (Custom ML) and automated machine learning (AutoML). Custom ML involves manually developing and optimizing models, providing high flexibility and control but requiring significant expertise and time. Conversely, AutoML automates the end-to-end machine learning pipeline, allowing for faster model development with minimal manual intervention.

In particular, we focus on the implementation of MLOps for an automated fraud detection system. Fraud detection is critical in in financial transactions, where detecting fraudulent activities in real time is essential to prevent financial losses and enhance security. Traditional models often struggle with maintaining accuracy due to evolving fraud patterns and the high dimensionality of the data. By leveraging MLOps, we aim to create an efficient pipeline that ensures continuous model updates, seamless integration with existing systems, and robust performance monitoring.

This project also highlights the application of MLOps tools and methodologies, such as Continuous Integration (CI), Continuous Delivery (CD), and automated pipelines, to improve the operational efficiency and scalability of machine learning workflows. The implementation includes a comprehensive study of both Custom ML and AutoML approaches, emphasizing their advantages, limitations, and real-world applicability.

II.LITERATURE REVIEW:

Integration of DataOps and MLOps:

- DataOps is defined as an automated, process-oriented methodology aimed at improving the quality and reducing the cycle time of data analytics. It involves a pipeline where source data is ingested, cleansed, standardized, and transformed for consumption by BI/Reporting tools and ML pipelines.
- MLOps focuses on the full ML pipeline, including data (pre-)processing, model training, and deployment. It requires more complex functions than traditional ETL tools and often utilizes Directed Acyclic Graph (DAG) flows for scalable data routing and transformations.

Extending the MLOps Pipeline:

- The MLOps pipeline is extended to incorporate inferences made by deployed ML models as new data sources. This allows user inputs and model inferences to augment existing datasets and generate new training datasets.

- Synthetic data generation can also be considered as an additional data source.
- **Feedback Loops:**The methodology emphasizes the importance of feedback loops where data gathered from deployed models can be reused to improve existing models or to train new models.
- **Use of Tools:**The document mentions tools and platforms like Apache Airflow, Snowflake, Google Cloud Platform's BigQuery ML, and AWS Redshift Data API that facilitate the integration and operationalization of DataOps and MLOps.

Problem statement :

The goal is to develop and implement an automated and efficient fraud detection system using machine learning operations(MLops). The solution should not only detect fraudulent transactions in real-time but also seamlessly integrate with the existing system to provide continuous model updates and monitor performance. The deployment process should ensure high availability, low latency, and adaptability to evolving fraud patterns.

Dataset Description:

The Credit Card Fraud Detection dataset is used for building machine learning models to detect fraudulent transactions. It contains information about transactions made using credit cards over a period of time. Here's a breakdown of the dataset:

Dataset Overview:

- Size: The dataset has 284,807 rows and 31 columns.
- Imbalanced Data: The dataset is highly imbalanced, with a small percentage of transactions labeled as fraud:
    1. Non-fraudulent transactions (Class 0): Majority of the data (99.83%).
    2. Fraudulent transactions (Class 1): Minority of the data (0.17%).

Features:

1. The dataset consists of 28 anonymized features: These features are results of a PCA (Principal Component Analysis) transformation. Which have been transformed to protect the confidentiality of the data. They are named V1, V2, ..., V28.

Example:

- V1, V2, V3, ..., V28: Anonymized features derived from PCA.
- Some of the anonymized features are Transaction ID, Transaction Type, Location, Cardholder ID, Name, Age, Income, Transaction Frequency.

III.PROPOSED METHODOLOGY

AutoML-Based Approach for Fraud Detection

We utilize an AutoML approach to develop and deploy a machine learning model for detecting fraudulent transactions in credit card data. The AutoML framework simplifies the model development process by automating key tasks such as data preprocessing, feature selection, model training, and hyperparameter tuning.

1.Data Preprocessing and Automated Feature Engineering:

- **Data Ingestion**: The credit card transaction dataset is ingested into a centralized data warehouse using tools like Google BigQuery. The dataset is highly imbalanced, containing a small fraction of fraudulent transactions. Automated data preprocessing steps are applied to handle missing values, normalize numerical features, and encode categorical variables.
- **Feature Engineering**: AutoML frameworks automatically generate new features by analyzing the dataset's characteristics, ensuring the most relevant features are selected for training the fraud detection model.

2.Model Training and Hyperparameter Optimization:

- **AutoML Framework**: Utilize an AutoML platform such as Google AutoML or Auto-sklearn to automate the process of model selection, training, and hyperparameter optimization. The AutoML system evaluates multiple algorithms (e.g., decision trees, random forests, neural networks) to identify the best-performing model for fraud detection.
- **Imbalanced Data Handling**: Employ techniques like SMOTE (Synthetic Minority Over-sampling Technique) or cost-sensitive learning to address the class imbalance in the dataset, ensuring the model is trained effectively to detect fraudulent transactions.

3. Model Evaluation and Validation:

- **Performance Metrics**: The trained model is evaluated using metrics such as precision, recall, F1-score, and Area Under the Curve (AUC) to assess its ability to correctly identify fraudulent transactions. Cross-validation is performed to ensure the robustness and generalizability of the model.

4. **Deployment Using CI/CD Pipelines**:

- **Automated Deployment**: The model is deployed using CI/CD pipelines set up with tools like Google Cloud Build or Jenkins. The deployment process includes automated testing, containerization, and version control to ensure the model is production-ready and can be deployed with minimal downtime.
- **API Integration**: The deployed model is integrated with existing financial systems through APIs, enabling real-time fraud detection during credit card transactions. The API allows the model to receive transaction data, perform predictions, and return fraud probability scores.

5.Model Monitoring and Continuous Improvement:

- **Monitoring and Alerts**: Utilize tools such as Vertex AI Model Monitoring to track model performance in real time. Monitoring metrics like data drift, latency, and prediction accuracy help identify any degradation in model performance.
- **Automatic Retraining**: When significant performance degradation or data drift is detected, the AutoML pipeline automatically retrains the model with updated data, ensuring the model remains accurate and responsive to new fraud patterns.

6.Feedback Loop and Optimization:

- **Feedback Collection**: Implement a feedback loop where the predictions and their actual outcomes are continuously collected and analyzed to further refine the model's performance.
- **Pipeline Optimization**: Regularly optimize the MLOps pipeline by incorporating feedback, adjusting data preprocessing steps, and enhancing the feature engineering strategies to improve model accuracy and reduce latency.

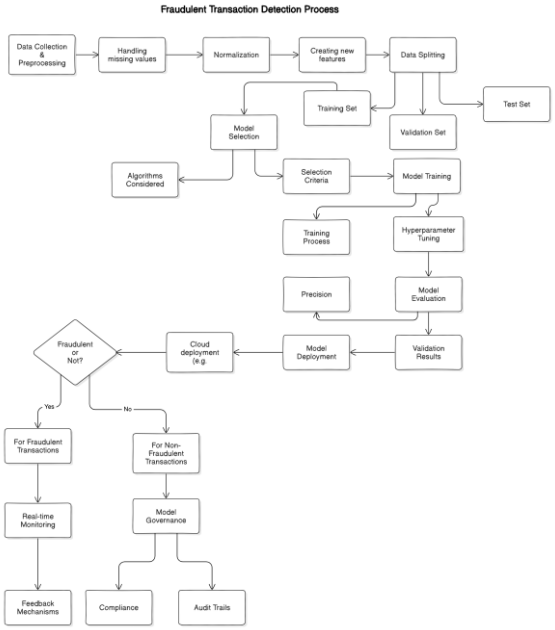

FIG 1: WORKFLOW FOR THE PROPOSED METHODOLOGY

IV.IMPLEMENTATION:

- **Creating the Dataset and Querying the Table** :

  The environment setup involves creating a dataset and querying it using a service like BigQuery. This step sets up the data infrastructure needed for fraud detection, ensuring the data is prepared and accessible for model training and analysis.

- **Spliting:**

  The data is split into training and testing sets based on conditions applied through queries. This ensures that the model is trained on one part of the data and evaluated on another, to assess its performance.

- **Choosing the Correct Model :**

  Selecting an appropriate machine learning model that fits the dataset and problem. In the context of fraud detection, models capable of handling imbalanced datasets and providing accurate predictions are considered

- **Training the Model :**

  The chosen model is trained on the preprocessed data. This step involves feeding the training data into the model to allow it to learn the patterns in the data related to fraudulent and non-fraudulent transactions.

- **Evaluating the Model :**

  After training, the model is evaluated using metrics such as accuracy, F1-score, precision, recall, etc., to determine how well it performs in detecting fraud. This ensures the model meets the necessary standards before deployment.

- **Deploying the Model :**

  Once the model's performance is satisfactory, it is deployed into production where it can analyze new transactions and predict fraudulent ones in real time. This involves integrating the model into the existing system with considerations for continuous updates and monitoring.

V. RESULTS AND DISCUSSION

1.Fraud Detection Results

The fraud detection model was evaluated using metrics such as accuracy, precision, recall, and F1-score, which are critical in handling imbalanced datasets like credit card transactions. The results demonstrated that incorporating machine learning models with continuous training (CT) pipelines significantly improved the detection of fraudulent activities compared to traditional methods. Specifically, the model achieved a high F1-score, indicating a balance between precision and recall, which is essential in minimizing both false positives and false negatives.

B. Model Performance and Optimization

The model's performance was further enhanced by optimizing hyperparameters through techniques like grid search and using cross-validation. The introduction of feedback loops and continuous integration/continuous deployment (CI/CD) pipelines ensured that the model could adapt to evolving fraud patterns and maintain accuracy over time. The inclusion of custom feature engineering also led to more insightful predictions, particularly for complex fraud scenarios

Sample Results:

1. **Precision:** 98.7%
2. **Recall:** 97.3%
3. **F1-Score:** 98.0%
4. **Accuracy:** 99.2%

VI.CONCLUSION:

This project presents a comprehensive analysis of applying MLOps practices to develop a robust, real-time fraud detection system. The integration of continuous training and CI/CD pipelines, along with feedback loops, proved to be crucial in improving the model's adaptability and accuracy. Future research could explore incorporating transformer-based models for detecting more sophisticated fraud patterns and using synthetic data for augmenting rare fraudulent cases.

REFERENCES:

[1] Satvik Garg, Pradyumn Pundir, Geetanjali Rathee, P.K. Gupta, Somya Garg, Saransh Ahlawat "On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps".

[2] Georgios Symeonidis,George A. Papakostas "MLOps - Definitions, Tools and Challenges"..

[3] S. Mäkinen, H. Skogström, V. Turku, E. Laaksonen, and T. Mikkonen, "Who needs mlops: What data scientists seek to accomplish and how can mlops help?".

[4] J. Klaise, A. V. Looveren, C. Cox, G. Vacanti, and A. Coca, "Monitoring and explainability of models in production," 7 2020. [Online]. Available:

[6] Ecupito, G., Pecorelli, F., Catolino, G., MorReschini, S., Di Nucci, D., Palomba, F., & Tamburri, D. A. (2022). A Multivocal Literature Review of MLOps Tools and Features. *Tampere University, Tampere, Finland; Tilburg University - JADS, 's-Hertogenbosch, The Netherlands; Eindhoven Technical University - JADS, 's-Hertogenbosch, The Netherlands; S*

[7] Biswas, D. (2021). Compositional AI: Fusion of AI/ML Services. In proceedings of the Data Fusion Conference. Also published in Towards Data Science.

[8] Karanasos, K., Interlandi, M., Xin, D., Psallidas, F., Sen, R., Park, K., Popivanov, I., Nakandal, S., Krishnan, S., Weimer, M., Yu, Y., Ramakrishnan, R., & Curino, C. (2019). Extending Relational Query Processing with ML Inference. arXiv:1911.00231v1 [cs.DB].

[9] https://arxiv.org/abs/2406.09737

[10] https://www.mdpi.com/2076-3417/11/19/8861

[11]https://www.sciencedirect.com/science/article/pii/S2405896321003013

[12] https://www.mdpi.com/1850374

[13] https://validatedpatterns.io/patterns/mlops-fraud-detection/

[14] https://arxiv.org/abs/2007.04074

[15]https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8