Propose a model monitoring pipeline and describe how you would track model drift.

1. Introduction

Machine learning (ML) model monitoring is a continuous process that tracks and evaluates ML models' behaviour and effectiveness in production environments [2]. To ensure model reliability and detect performance degradation, a robust monitoring pipeline is essential.

2. Proposed Model Monitoring Pipeline Design

Refer to Figure 1 for the proposed pipeline design.

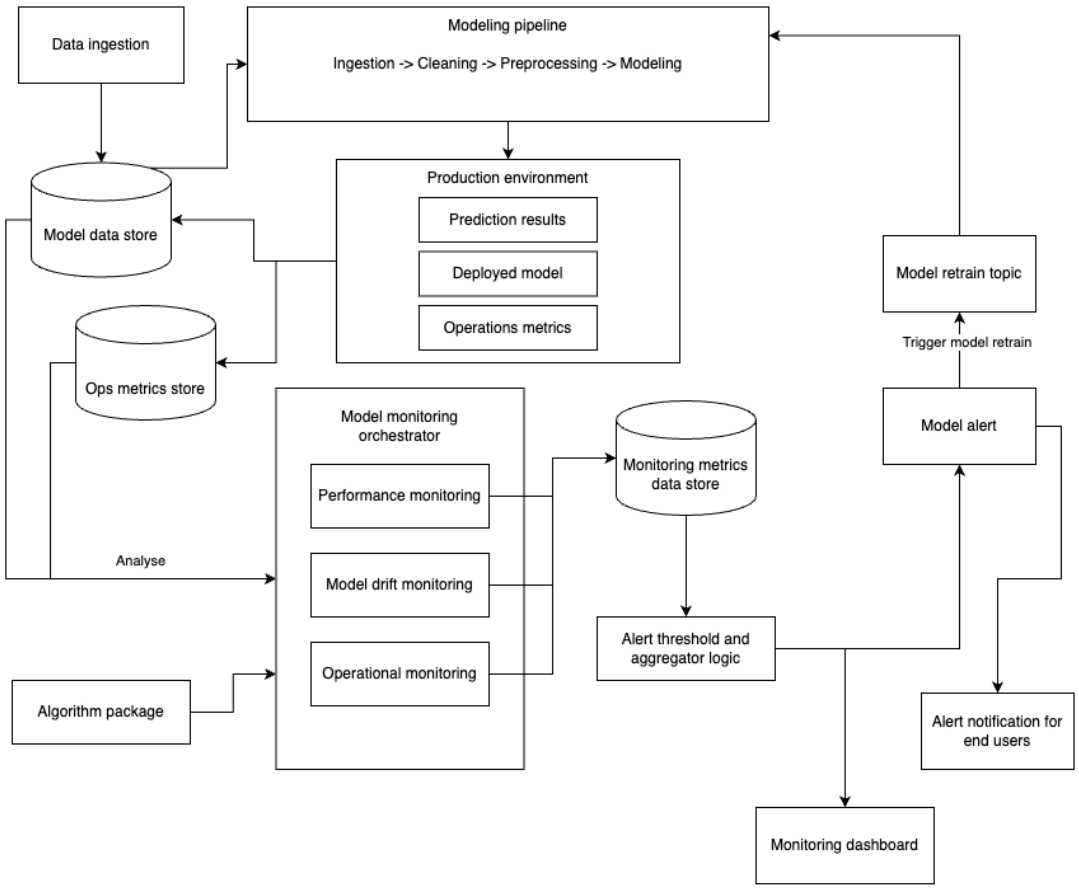

Figure 1: Proposed design for model monitoring pipeline

Data Ingestion

The Data Ingestion component manages continuous data flow into the system, with tools like AWS Kinesis handling the streaming data processing.

Data Store

- **Model Data Store:** Models, training data, and predictions
- **Monitoring Metrics Store:** Drift and performance metrics, together with analytical results from the monitoring system.
- **Ops Metrics Store:** Latency, throughput, and resource usage metrics.

Possible storage options that can be used include Amazon S3, while incorporating Delta Lake to ensure ACID compliance and data reliability [7].

Application Orchestrator

The Application Orchestrator serves as the core processing engine, executing monitoring logic through periodic batch checks on a scheduled cadence. It can run on distributed compute clusters, where it implements monitoring across key metric areas using technologies like PySpark and SciPy. The Algorithm Package isolates metric computation logic from infrastructure code.

- **Model performance monitoring:** Tracks metrics including accuracy, precision-recall curves for classification models, while regression models are evaluated using RMSE and MAE [3].
- **Model drift monitoring:** Utilizes algorithms like Kolmogorov-Smirnov test and Jensen-Shannon divergence to identify concept drift and distribution changes [4].
- **Operational monitoring:** Tracks system health through metrics like latency, throughput, and resource utilization.

Major platforms such as Amazon SageMaker [4] can be considered as solutions for these monitoring tasks, and specialized tools like Evidently AI can be used for feature drift detection.

Automated Alert

The Automated Alert system provides alerting based on predefined thresholds, utilizing tools like PagerDuty and email systems to notify stakeholders when major model degradation is detected.

Feedback Loop and Retraining Trigger

The Feedback Loop and Retraining Trigger component automates the model retraining process when performance degrades below acceptable levels. Tools like MLflow and Apache Airflow can manage these automated workflows.

Monitoring Dashboard

The Monitoring Dashboard provides visualization of all monitored metrics and data. It typically employs Prometheus for metrics collection, with visualization tools like Grafana for creating displays that allows for trend and potential issues identification.

3. Model Drift Detection Methods

Model drift occurs when a ML model's performance deteriorates over time as real world data patterns deviate from those used during training. These methods can be used to monitor different aspects of the model drift:

- **Population Stability Index:** This helps to quantify the difference between score distributions. Values of more than 0.2 indicates significant drift and need investigation.
- **Kolmogorov-Smirnov Test:** This test helps to compare prediction distributions [1, 4, 5].

Model drift is often caused by concept drift [6]. Concept drift represents fundamental changes in the relationships between input features and target outputs. Detection typically requires combining multiple statistical approaches to monitor both feature distributions and model quality metrics like recall and precision [1, 6].

4. Conclusion

Ultimately, a comprehensive model monitoring pipeline is important for maintaining model reliability and effectiveness in dynamic, real-world environments.

References

[1] Datadog, “Machine learning model monitoring: Best practices,” Datadog, https://www.datadoghq.com/blog/ml-model-monitoring-in-production-best-practices (accessed Feb. 9, 2025).

[2] “Model monitoring for ML IN PRODUCTION: A comprehensive guide,” Evidently AI - Open-Source ML Monitoring and Observability, https://www.evidentlyai.com/ml-in-production/model-monitoring#model-monitoring-strategy (accessed Feb. 9, 2025).

[3] “A Guide to Monitoring Machine Learning Models in production,” NVIDIA Technical Blog, https://developer.nvidia.com/blog/a-guide-to-monitoring-machine-learning-models-in-production/ (accessed Feb. 9, 2025).

[4] B. Eck, *A monitoring framework for deployed machine learning models with supply chain examples*, Nov. 2022.

[5] “Model drift: Best practices to improve ML model performance,” Encord, https://encord.com/blog/model-drift-best-practices/#:~:text=You%20can%20detect%20model%20drift,across%20training%20and%20production%20workflows. (accessed Feb. 9, 2025).

[6] “What is concept drift in ML, and how to detect and address it,” Evidently AI - Open-Source ML Monitoring and Observability, https://www.evidentlyai.com/ml-in-production/concept-drift#concept-drift-vs-model-drift (accessed Feb. 9, 2025).

[7] “Delta Lake on S3,” Delta Lake, https://delta.io/blog/delta-lake-s3/ (accessed Feb. 9, 2025).