

PREDICTIVE MAINTENANCE ON ELECTRIC VEHICLE TELEMETRY DATA USING AN END-TO-END MLOPS PIPELINE

Nguyen Ngoc Tam, Bui Duc Lan, Nguyen Anh Duc, Nguyen Ba Son

¹ FPT University / FPT School of Business & Technology (FSB), Vietnam

Email: nguyenngoctam1307@gmail.com, buiduclan000@gmail.com, aduc8386@gmail.com,
sonnb.ktpm@gmail.com

Abstract:

Predictive maintenance plays a critical role in improving reliability and reducing operational costs in electric vehicle (EV) fleets. This project presents an end-to-end MLOps system designed to predict potential vehicle failures using real-world EV telemetry data. The proposed system integrates data ingestion, model training, deployment, monitoring, and continuous integration using modern DevOps, DataOps, and MLOps practices. By leveraging Kafka, MLflow, Docker, FastAPI, and Prometheus, the system ensures scalability, reproducibility, and observability across the entire machine learning lifecycle.

Keywords: Predictive Maintenance, MLOps, DataOps, DevOps, Electric Vehicles, Kafka, MLflow, XGBoost

I. INTRODUCTION

Electric vehicles generate large volumes of telemetry and sensor data during operation. Traditional maintenance strategies rely on reactive or scheduled maintenance, which may lead to unnecessary costs or unexpected failures. Predictive maintenance addresses this challenge by forecasting failures before they occur using machine learning models.

However, deploying machine learning models in production introduces additional complexity related to data pipelines, model versioning, deployment automation, and monitoring. This project aims to solve these challenges by designing a complete MLOps pipeline for predictive maintenance in EV fleets.

II. PROBLEM STATEMENT

The main challenges addressed in this project include:

- Continuous ingestion of high-frequency IoT telemetry data
- Reliable preprocessing and feature extraction
- Reproducible model training and versioning
- Scalable model deployment and inference
- Monitoring model performance and system health
- Automated CI/CD for ML services

The goal is to build a production-ready predictive maintenance system rather than a standalone machine learning model.

III. DATASET DESCRIPTION

The project uses the EVIoT-PredictiveMaint Dataset, a real-world dataset collected from IoT-enabled electric vehicles over a 5-year period (2020–2025).

Dataset characteristics:

- 175,393 records
- 15-minute sampling interval
- Multi-modal telemetry (battery, motor, temperature, usage)
- Environmental conditions
- Historical maintenance events

This dataset supports multi-horizon failure prediction and is suitable for fleet-scale predictive maintenance research.

IV. SYSTEM ARCHITECTURE

The system follows a microservices-based MLOps architecture as shown below:

Core Components:

- Data Ingestion: Apache Kafka
- Data Processing & Training: Python + XGBoost
- Experiment Tracking: MLflow
- Model Storage: MinIO (S3-compatible storage)
- Inference API: FastAPI
- Containerization: Docker & Docker Compose
- CI/CD: GitLab CI/CD
- Monitoring: Prometheus & Grafana
- Version Control: GitLab

Architectural Principles:

- Decoupled services
- Containerized workloads
- Infrastructure as Code mindset
- Observability-first design

V. SYSTEM WORKFLOW

The end-to-end workflow of the system is described as follows:

- Data Ingestion: EV telemetry data is streamed or batch-ingested via Kafka.
- Data Processing: Python-based preprocessing pipelines clean, transform, and prepare features for training.
- Model Training: XGBoost models are trained using processed data. Experiments, parameters, and metrics are tracked using MLflow.
- Artifact Storage: Trained models and datasets are stored in MinIO with versioning support.
- Model Deployment: The trained model is deployed as a FastAPI inference service inside a Docker container.
- Monitoring: Prometheus collects system and inference metrics, while Grafana visualizes performance and alerts.
- CI/CD Automation: GitLab CI/CD automates testing, building, and deployment of services.

VI. MLOPS, DATAOPS, AND DEVOPS ALIGNMENT

This project aligns with core concepts from the course:

- DevOps: CI/CD pipelines, Docker-based deployments, infrastructure automation
- DataOps: Automated data pipelines, data versioning, reproducibility
- MLOps: Model lifecycle management, experiment tracking, monitoring, and retraining readiness

The system reflects a production-grade MLOps maturity model, emphasizing reliability and scalability.

VII. EVALUATION AND MONITORING

The system monitors:

- Model inference latency
- Prediction frequency
- Service availability
- Data drift indicators (extensible)

Grafana dashboards provide real-time insights, enabling rapid diagnosis and system observability.

VIII. CONCLUSION AND FUTURE WORK

This project demonstrates how predictive maintenance can be operationalized using modern MLOps practices. By integrating data pipelines, model lifecycle management, and monitoring, the system moves beyond experimental machine learning toward production readiness.

Future improvements include:

- Online learning and automated retraining
- Advanced data drift detection
- Federated learning across multiple EV fleets
- Integration with cloud-native orchestration (Kubernetes)

IX. REFERENCES

1. Duke University, DevOps, DataOps, and MLOps, Coursera, 2023.

Available: <https://www.coursera.org/learn/devops-dataops-mlops-duke/>

2. D. Sculley et al., “Hidden Technical Debt in Machine Learning Systems,” Advances in Neural Information Processing Systems (NeurIPS), 2015.
3. MLflow, “MLflow: An Open Source Platform for the Machine Learning Lifecycle,” <https://mlflow.org>.
4. Prometheus, “Prometheus Monitoring System,” <https://prometheus.io>.
5. T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.