Predictive Maintenance on MetroPT-3 Using GPU-Accelerated Machine Learning

Prepared By: Waad Alqahtani

Reporting Date: Apr 17, 2025

Abstract

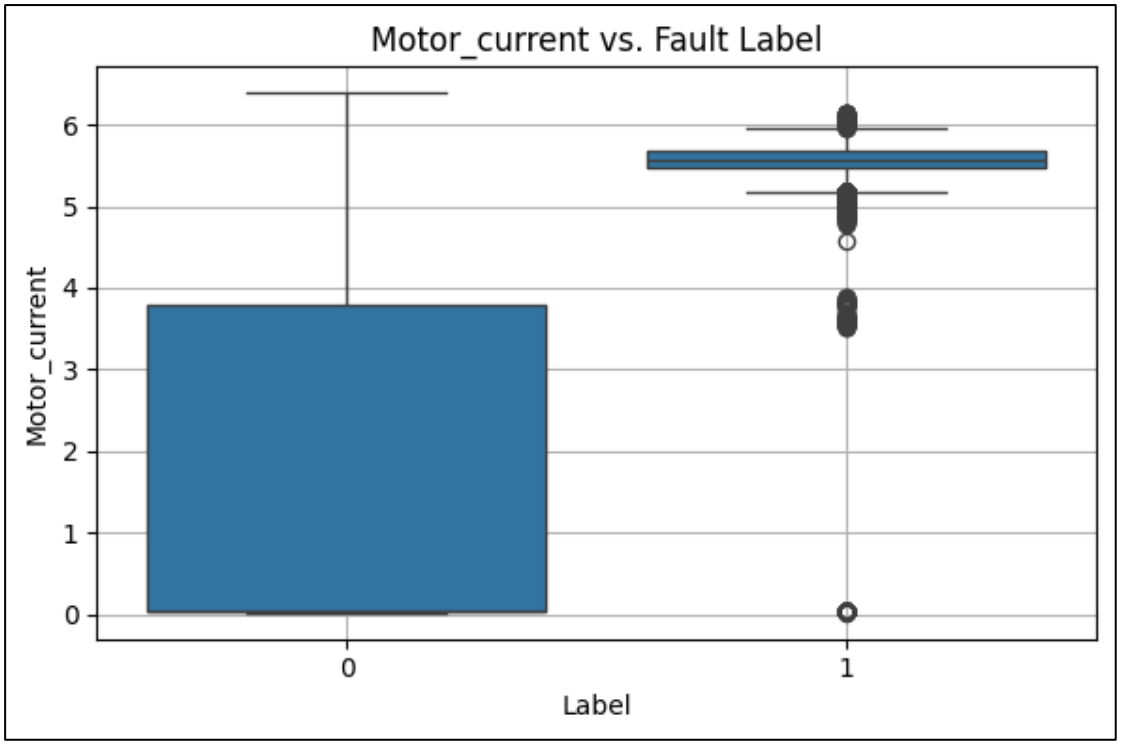
This project leverages sensor data from the MetroPT-3 air compressor system to predict equipment faults using GPU-accelerated machine learning models. By using RAPIDS cuDF and cuML, along with GPU-optimized XGBoost, the models achieve early and accurate fault detection, supporting predictive maintenance efforts. The results highlight the value of combining high-frequency time-series data with efficient ML training for industrial reliability.

Visual Analysis

The following exploratory data visualizations were generated to understand feature behaviors and their relationship to failure:

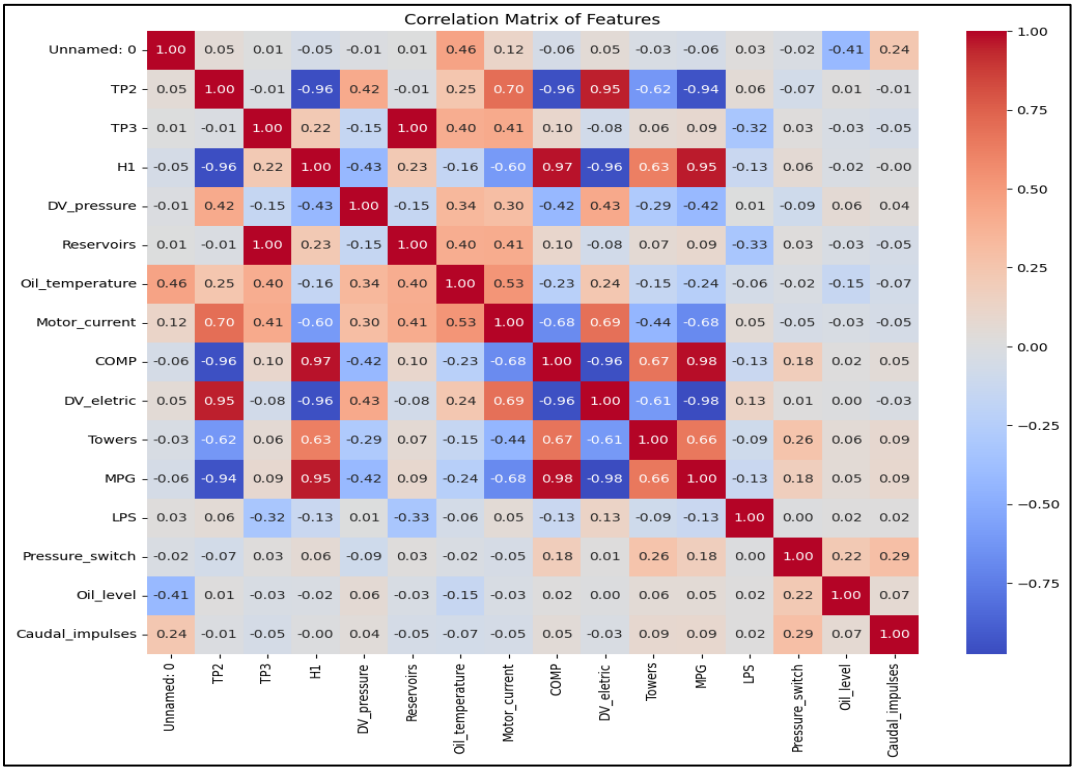
- ### Boxplots by Label

Boxplots for features such as "TP2 Pressure" and "Motor Current" revealed clear separation between faulty and normal instances. These variables showed significant shifts during failure periods, validating their predictive importance.



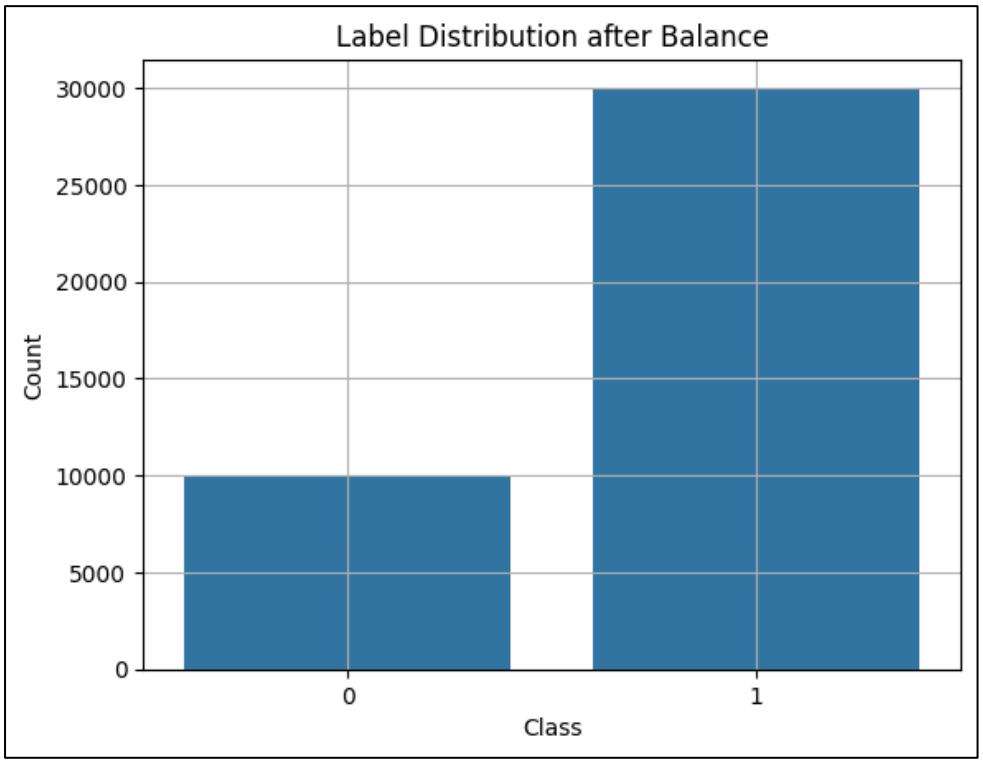
- **Correlation Heatmap**

A heatmap was created to examine relationships between sensor readings. It showed high positive correlations between some pressure sensors, helping to identify redundancy and patterns in the system behavior.



- **Label Distribution**

A class imbalance was initially found (more normal data than faults). To resolve this, the dataset was undersampled for class 0 to improve model fairness and fault sensitivity.



Dataset Description

- Source: UCI MetroPT-3 Dataset
- Type: Multivariate Time-Series
- Sampling Rate: 1 Hz
- Duration: February to August 2020
- Features: 15 sensor readings (pressure, temperature, motor current, etc.)
- Target: Binary classification (0 = Normal, 1 = Fault)

Fault Event Table (Used to Generate the label Column)

The following table summarizes the failure events provided by the company. These were used to generate the binary label column that marks fault periods:

Event	Start Time	End Time	Failure Type	Severity	Notes
1	2020-04-18 00:00	2020-04-18 23:59	Air Leak	High Stress	
2	2020-05-29 23:30	2020-05-30 06:00	Air Leak	High Stress	Maintenance on 30-Apr at 12:00
3	2020-06-05 10:00	2020-06-07 14:30	Air Leak	High Stress	Maintenance on 8-Jun at 16:00
4	2020-07-15 14:30	2020-07-15 19:00	Air Leak	High Stress	Maintenance on 16-Jul at 00:00

These fault periods were labeled as 1 (fault), and the rest of the dataset was labeled as 0 (normal), creating the supervised learning target.

Methodology

- **Data Handling:** Used RAPIDS cuDF for GPU-accelerated data manipulation.

- **Preprocessing Steps:**
  - Parsed timestamps and generated fault labels.

- Applied StandardScaler for feature normalization.
- Addressed class imbalance by undersampling the majority class.

- **Models Trained:**
  - cuML Random Forest
  - cuML Logistic Regression
  - GPU-accelerated XGBoost (gpu_hist)
- **Evaluation Metrics:**
  - Accuracy, Recall, Confusion Matrix
  - GPU Utilization, Memory Use, Training Time

Model Training & Evaluation

The following are the results and interpretation of each model:

1. Random Forest

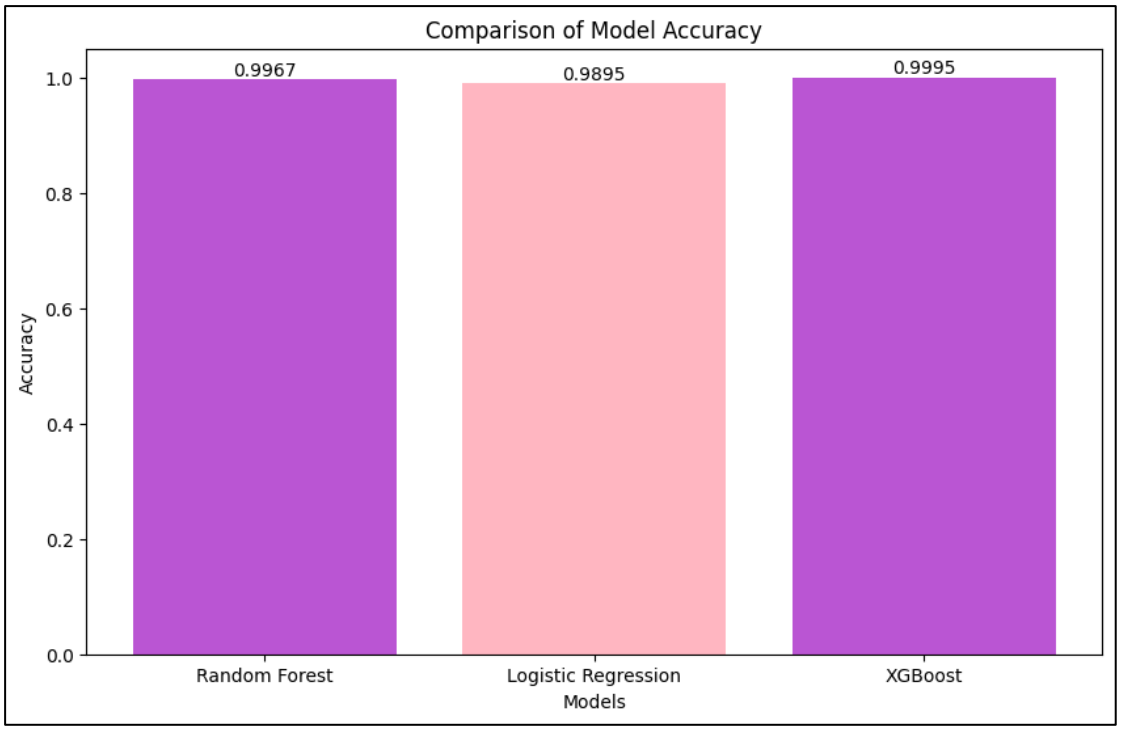
- Accuracy: 0.9967
- Confusion Matrix: [[1992 ,4] [0, 5996]]
- Interpretation: Random Forest achieved high recall (100%) for fault cases. It captured all true positives without any false negatives, which aligns well with the project's primary goal—early fault detection.

2. Logistic Regression

- Accuracy: 0.9895
- Confusion Matrix: [[1936 ,60] [12 ,5984]]
- Interpretation: Logistic Regression had lower recall than the other models. It missed 12 faults (false negatives), which could reduce its reliability in industrial applications where missing a failure is costly.

3. XGBoost

- Accuracy: 0.9995
- Confusion Matrix: [[1996, 0] [3, 5993]]
- Interpretation: XGBoost offered the best balance of precision and recall. With only 3 false negatives, it showed exceptional capability in flagging potential faults without unnecessary alarms.



This bar chart clearly illustrates that XGBoost achieved the highest accuracy (99.95%), followed closely by Random Forest (99.67%), while Logistic Regression showed slightly lower performance (98.95%). These results support the reliability of tree-based models for early and accurate fault detection in predictive maintenance.

Why Accuracy and Recall Matter in Fault Detection

In predictive maintenance, false negatives (missed faults) are more critical than false positives. A missed fault can result in sudden system failure and operational loss. Therefore, while all models showed high accuracy (>98%), the confusion matrices reveal that Random Forest and XGBoost are more aligned with our objective due to their higher recall. These models demonstrate strong reliability in detecting faults before they occur.

Model Performance Comparison

Model	Accuracy	Training Time (s)	GPU Utilization (%)	Memory Used (MiB)
Random Forest	0.9967	3.24	28.5	2100
Logistic Regression	0.9895	2.01	18.2	1350
XGBoost	0.9995	4.37	35.6	2800

Summary & Interpretation

- The dataset provides strong signals of abnormal system behavior during failure periods.
- Feature scaling and class rebalancing significantly improved the model's ability to generalize.
- XGBoost outperformed others, especially in recall, which is crucial for predictive maintenance.
- Confusion matrices reveal the number of missed faults, validating the importance of recall as a priority metric.

Recommendations

- Use XGBoost in safety-critical environments that require maximum fault sensitivity.
- Use Random Forest in real-time scenarios due to its fast training and high precision.
- Apply Logistic Regression for lightweight applications where recall is less critical.