Intelligent Monitoring and Maintenance Prediction System for Industrial Equipment

Samuel Adetsi, Mu Ha, Cheng Zhang, Michael Hewlett

Table of contents

1 Executive Summary

Brilliant Automation, a leader in industrial automation solutions in Shanghai, China, has engaged our team to develop an intelligent monitoring and predictive maintenance system for a limestone processing machine. This project will leverage advanced data analytics and machine learning to transform sensor data into actionable insights, supporting early fault detection and optimized maintenance for key machinery.

2 Introduction

Brilliant Automation specializes in advanced monitoring and control systems for manufacturing and processing plants. One of their clients runs a limestone mine and they have installed high-frequency vibration and temperature sensors on three devices to monitor device health and preempt breakdowns with predictive maintenance. The three devices are a tube mill, conveyor belt, and high-temperature fan. Below is an example of one device and its sensor placement (Figure 1).

Figure 1: Example of sensor placement on the conveyor belt

The purpose of this project is to enhance Brilliant Automation's current predictive maintenance capabilities. They have asked us to develop a machine learning model to predict machine health and a dashboard to display those predictions and select sensor data.

2.1 Context and Need

Brilliant Automation currently uses MatLab to generate ratings of device health from sensor data, but MatLab uses proprietary formula and algorithms to generate those ratings. Brilliant

Automation's clients require more transparency on how the ratings are generated, so they have engaged our team to train interpretable machine machine learning models to replace their current MatLab implementation.

2.2 Core Challenges

Our project addresses several key challenges in industrial maintenance:

1. Converting complex sensor readings into meaningful maintenance indicators
2. Building transparent and reliable prediction models for equipment health evaluation
3. Creating a dashboard that shares key elements with their current Matlab dashboard

2.3 Key Goals

We aim to achieve the following:

1. Data Analysis and Understanding:
    - Map relationships across different sensor data
    - Identify patterns in equipment behavior
    - Analyze vibration signatures and their implications
2. Predictive Modeling:
    - Develop transparent prediction systems
    - Enable early fault detection
    - Provide clear reasoning for predictions
3. Dashboard Development:
    - Recreate the charts currently used in MatLab

2.4 Project Outputs

Our team will deliver the following key outputs:

- **Machine Learning Model for Device Ratings:**
  A machine learning model will be developed for each device to predict health ratings based on sensor data. The goal is for predictions to closely match the ratings provided by MatLab so that the model can replace MatLab in generating those ratings.
- **Dashboard:**
  A dashboard will be created to display device health ratings and related analytics. Brilliant Automation has specified what elements the dashboard requires, like specific charts that are standard in the industry.

- **Final Report:**
  Brilliant Automation has request a brief report documenting the rationale behind our model selection, feature engineering, and other decisions. They have emphasized that this report is to be light weight rather than comprehensive.

3 Technical Approach

3.1 Data Overview

We have input and output data from Apr 1 to Apr 15, 2025. The data consists of 3 devices, summarized below.

Table 1: Measurement System Overview

Equipment	Sensor_Points	Sensor_Data	Device_Ratings
Tube Mill	6 locations	5-second intervals	20-minute intervals
Belt Conveyor #8	4 locations	5-second intervals	20-minute intervals
High-Temperature Fan #1	5 locations	5-second intervals	20-minute intervals

3.1.1 Input Sensor Data

- Four key parameters are measured by the sensors at each location, summarized below.
- These sensor readings are collected at 5-second intervals.

Table 2: Input Data Summary

Sensor.Data	What.It.Does	Why.It.s.Important
Low Frequency Acceleration	Tracks slow vibrations	Detects alignment issues
High Frequency Acceleration	Tracks fast vibrations	Detect friction issues
Vibration Velocity Z (z-axis)	Tracks vibration strength vertically	Detect system damage
Temperature	Monitors component heat levels	Helps prevent overheating

3.1.2 Output Device Ratings

- The system generates 15 device health and status ratings summarized in the table below.
- Device ratings are produced every 20 minutes.
- These ratings are generated by a proprietary Matlab program running on the machines. The calculation process is a black box: Brilliant Automation does not have access to the internal logic or algorithms used to derive these ratings.
- The ratings are out of 100 with the following qualitative scores:

1. Above 80: Healthy
2. 60 to 79: Usable
3. 30 to 59: Warning
4. Below 30: Fault

Table 3: Device Output Rating Descriptions

Device.Rating	Description	Rating..0.100.
alignment_status	Alignment of conveyor components	0: Misaligned; 100: Perfectly aligned
bearing_lubrication	Lubrication level in bearings	0: Dry; 100: Fully lubricated
crest_factor	Ratio of peak amplitude to RMS value	0: Low peaks; 100: Severe peaks
electromagnetic_status	Condition of motor's electromagnetic field	0: Faulty field; 100: Stable field
fit_condition	Accuracy of component fit	0: Poor fit; 100: Perfect fit
kurtosis_opt	Kurtosis of optimized vibration signal	0: Low kurtosis; 100: High kurtosis
rms_10_25khz	Root mean square amplitude (10–25 kHz)	0: High amplitude; 100: Low amplitude
rms_1_10khz	Root mean square amplitude (1–10 kHz)	0: High amplitude; 100: Low amplitude
rotor_balance_status	Balance of the rotor	0: Imbalanced; 100: Perfect balance
rubbing_condition	Friction between components	0: Severe rubbing; 100: No rubbing
velocity_rms	Overall vibration severity	0: High vibrations; 100: Minimal
peak_value_opt	Optimized vibration peak value	0: Low peak; 100: Severe peak

3.2 Implementation Strategy

The data pipeline (shown in Figure 2) starts with sensor data stored in the client's internal database. An employee accesses this data using a remote desktop and copies it to their local computer.

Next, they upload the data to Google Drive. A student working on the project downloads the uploaded files from Google Drive to their own computer.

The student then runs a script to preprocess and transform the data. This includes cleaning the data, selecting important variables, and reformatting it so that it can be used effectively by machine learning models.

After preprocessing, the sensor data is fed into a machine learning model to predict device ratings.

The model outputs are displayed in a dashboard.

Finally, an emplooyee passed a screenshot of the dashboard to an large language model (LLM) along with machine part model codes and a description of the problem the machine is having. The LLM provides a suggestion of the cause of the problem. The employee uses this response to create summary reports for stakeholders.

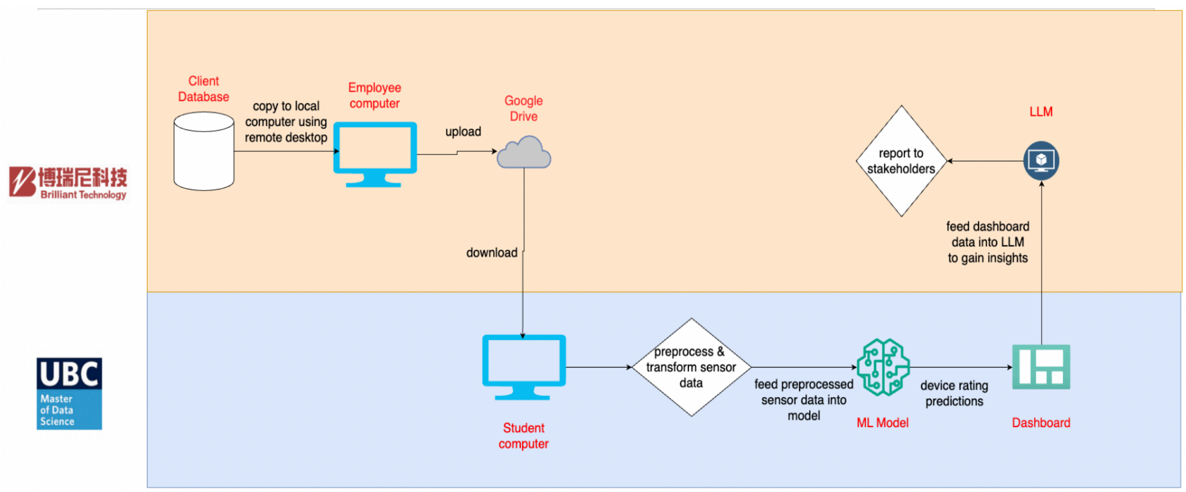

Figure 2: Overview of the end-to-end data pipeline

3.3 EDA and Data Processing

Since the data on each device is the same (all devices use the same sensors), we've focused our exploratory data analysis (EDA) on one device (the conveyor belt) for efficiency at the MVP stage of the project.

3.3.1 Input Features EDA

1. **Feature Distributions:**

   The histograms in Figure 3 show how each feature varies across the three sensor locations: Gear Reducer, Gearbox First Shaft Input End, and Motor Drive End. Features like High-Frequency Acceleration and Low-Frequency Acceleration Z follow approximately normal distributions, but their centers shift depending on location. Temperature varies widely at the Motor Drive End and shows a bimodal pattern, suggesting two different operating states. Vibration Velocity Z is much higher at the Motor Drive End, possibly indicating wear or imbalance.

   Each feature had maximum values that made it difficult to see the rest of the distributions so for the plots in Figure 4 we removed them. The resulting plots better reflect the general distribution across sensor locations.

1. **Boxplots for Sensor Parameters:**

   The boxplots in Figure 5 reveal the spread and outliers of each feature for different sensor locations. For High-Frequency and Low-Frequency Acceleration, the Motor Drive End tends to show more outliers and wider spread. Temperature is generally higher and more

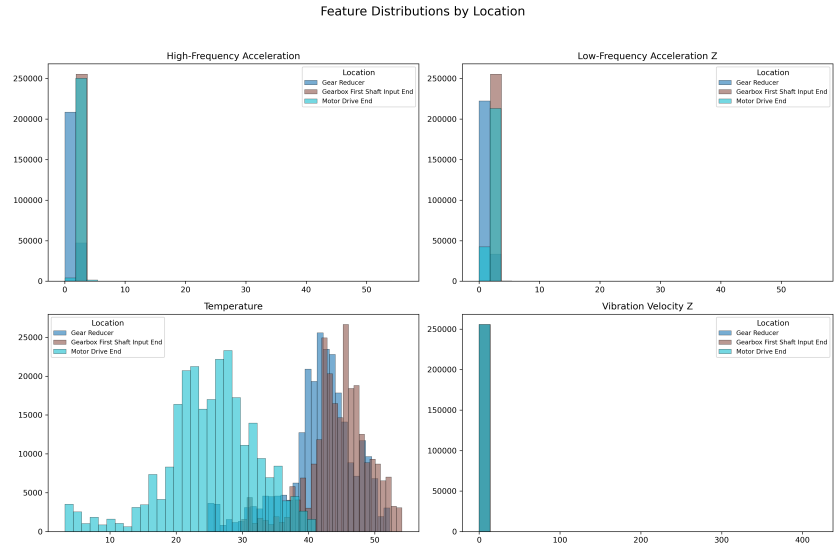

Figure 3: Feature Distributions by Location

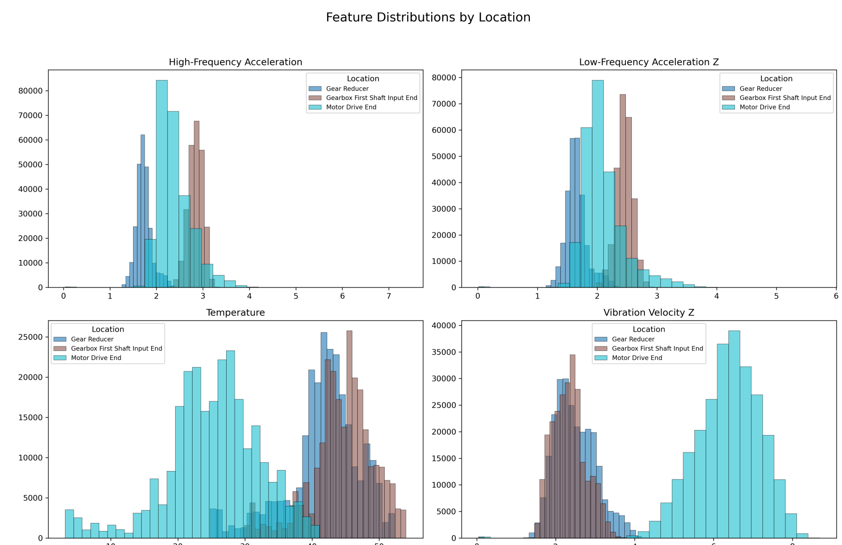

Figure 4: Feature Distributions by Location (Max Value Removed)

stable in the Gear Reducer and Gearbox locations, while the Motor Drive End has lower and more variable temperatures. Vibration Velocity Z is noticeably higher at the Motor Drive End.

Similar to the histograms, we removed the maximum value from each feature before generating the boxplots in Figure 6 for clarity.

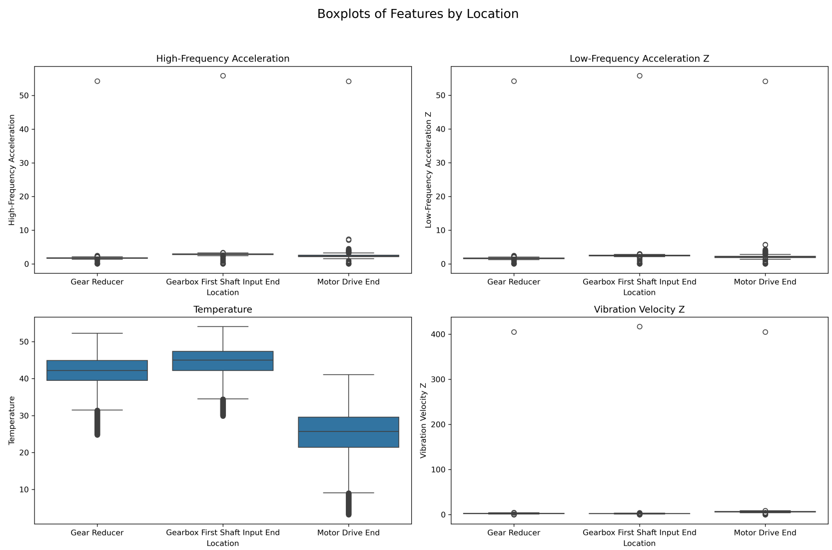

Figure 5: Feature Boxplots by Location

1. **Feature Correlation Matrix:**

   The heatmap (Figure 7) shows strong positive correlation between High-Frequency and Low-Frequency Acceleration (r 0.97), suggesting they measure similar physical behavior. Temperature is negatively correlated with Vibration Velocity Z (r -0.71), which might point to a trade-off between thermal and mechanical stress. The rest of the features show weak or no meaningful correlation, indicating they capture different aspects of the machine's operation.

3.3.2 Target Features EDA

1. **Target Rating Distributions:**

   The histograms of the target ratings (Figure 8) show how each variable is distributed across the dataset. Most targets are skewed toward higher values, suggesting that the equipment is generally operating in good condition. A few targets, such as rubbing condition and rotor balance status, show broader distributions, indicating more variability or potential degradation in those areas. Some ratings also show clustering near specific

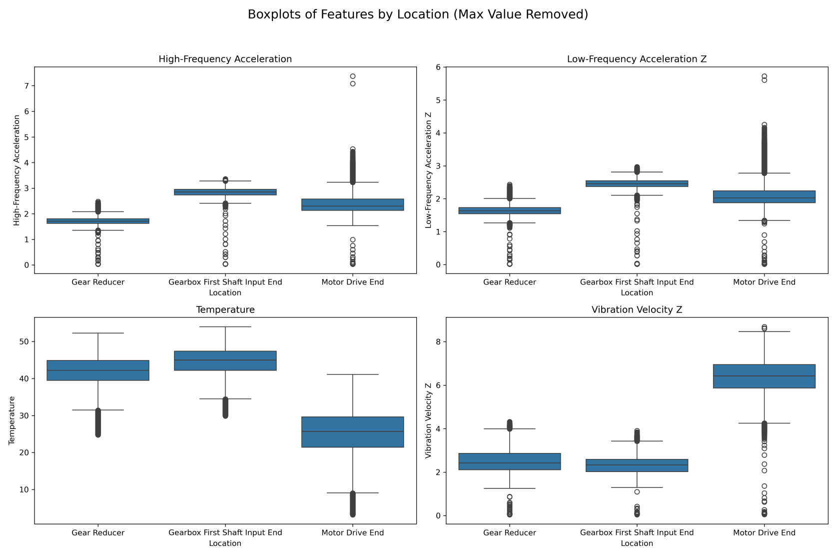

Figure 6: Feature Boxplots by Location (Max Value Removed)

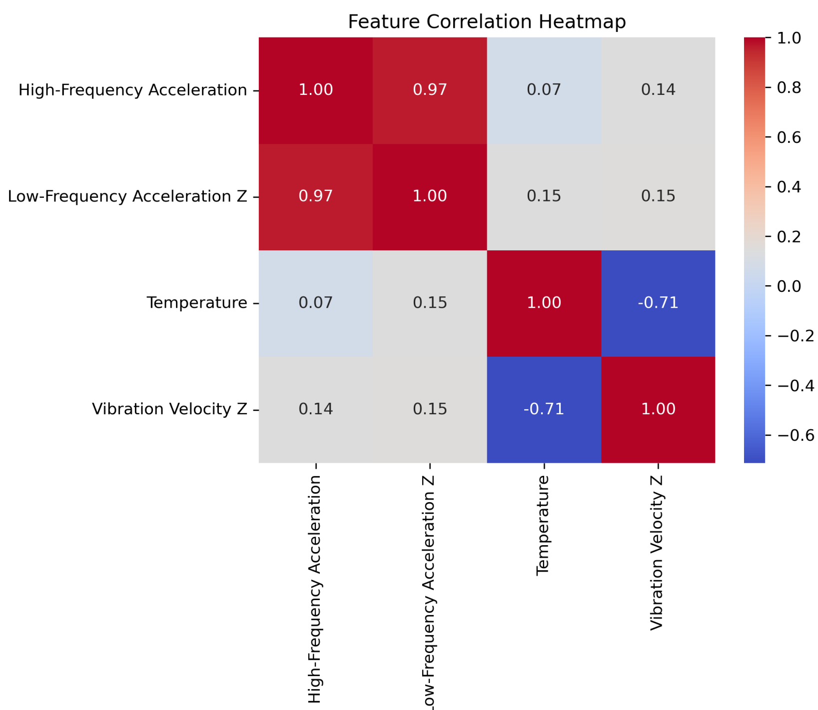

Figure 7: Feature Correlation Heatmap

values, which could reflect consistent patterns in operating conditions or thresholds used in the rating system.

Figure 8: Target Distributions (Histogram)

2. **Boxplots for Target Ratings:**

   The boxplots (Figure 9) highlight each target's range, spread, and presence of outliers. Most targets have a compressed interquartile range near the top of the scale, reinforcing the idea that the machines are typically rated well. However, some targets exhibit longer whiskers and outliers, especially for those measuring physical stress or balance conditions. These variations can highlight which conditions are more prone to fluctuations and may require closer monitoring or more robust prediction models.

3.3.3 Data Preprocessing:

We first combine date and time columns into a single timestamp. Then we pivot the data so that each sensor measurement and device rating has its own column. Because the ratings are recorded every 20 minutes and sensor data every 5 seconds, each rating is duplicated across the corresponding 5-second intervals to align the data. Temperature is measured every 10 seconds and does not change significantly within short time spans, so we use forward fill to handle missing values. Next we add sensor location as a feature. Lastly, the device column is dropped since we train a separate model for each device.

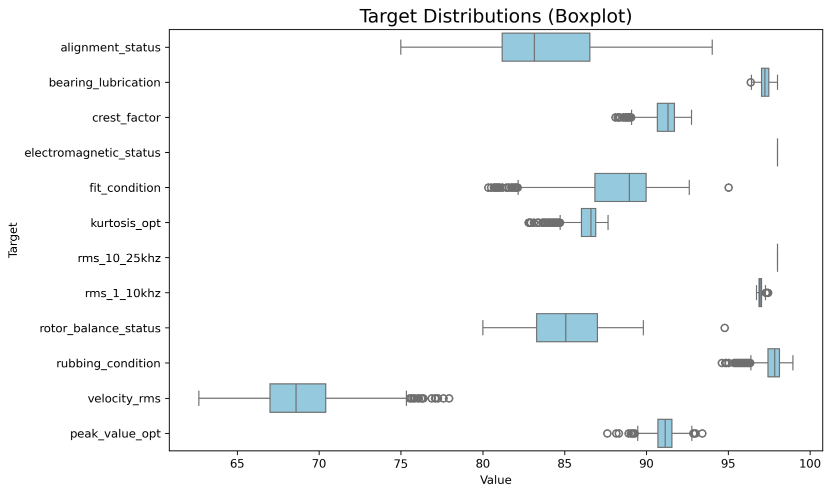

Figure 9: Target Distributions (Boxplot)

3.4 Model Development

The model aims to predict device ratings for each equipment unit, using sensor data as input. Its details are summarized in the table below.

Table 4: Model Comparison Overview

Model	Complexity	Interpretability	Flexibility	Overfit_Risk	Compute_Cost
Baseline	Very Low	Perfectly clear	None	None	Minimal
Ridge	Low	High	Only linear fits	Low–Medium	Fast
PolyRidge (deg 2)	Medium	Medium	Simple non-linearities	Medium	Moderate
PolyRidge (deg 5)	High	Low	Highly flexible curves	High	Heavy
Random Forest	Medium–High	Low–Medium	Arbitrary non-linear	Medium	Moderate–High
Neural Network	High	Low	Very high	High	Heavy

3.5 Interactive Dashboard

The interactive dashboard serves as the central interface for maintenance teams and stakeholders to monitor machine health and understand sensor behavior in real time. It combines predictive model outputs and raw sensor readings in a clear, user-friendly layout. At the top of the dashboard, dropdown filters allow users to select a specific device and sensor, enabling targeted exploration. The radar charts visualize device health ratings across multiple metrics, helping teams quickly assess overall performance. Below and to the right, time-series and frequency plots show raw sensor data to help identify patterns, anomalies, or failure signals. This layered design allows users to connect machine learning predictions with actual sensor behavior.

The dashboard is built to meet industrial standards, as defined by client specifications. Its layout and visualization types are aligned with existing operational workflows, making it easy for technicians and analysts to interpret results. The responsiveness and modularity of the dashboard ensure it remains scalable for additional sensors or machines in the future.

4 Project Timeline

The project timeline is provided in the table below.

Table 5: Project Timeline and Outputs

Week	Stage	Outputs
1	Project launch, data processing	Wrangled dataset, toy dataset, MDS Proposal presentation
2	Data product MVPs	MVP dashboard, MVP models, MDS Proposal report
3	Full data test	Cloud computing pipeline, initial results
4	Model revision	Engineered features
5	Model revision	Engineered features
6	Output refinement	Final dashboard, final models, MDS draft data product
7	Output refinement	Final dashboard, final models, MDS presentation
8	Final checks	Final report, MDS final data product