Understanding, Detecting, and Predicting Machine Failures Using Machine Learning Models

Yumeng Pan, Minghao Li, Yilin Shi, YungTing Lu, and Xinyu Yang

Delft University of Technology, 2500 AA Delft, the Netherlands

Abstract. This study applied machine learning (ML) to understand and predict failures using operational data from 205 machines, addressing severe class imbalance (0.14% failures). K-means clustering identified 7 distinct failure modes associated with unique sensor patterns and operational contexts. Logistic Regression, utilizing SMOTE-balanced data, successfully differentiated failure from normal states (Recall=0.71), confirming data utility despite low precision and outperforming a baseline linear model. For predicting Time To Failure, Random Forest yielded the highest accuracy by standard metrics (R^2=0.8585), while Logistic Regression excelled at providing timely early warnings (100% accuracy within a 30-day window). This research demonstrates ML's effectiveness for predictive maintenance, highlighting Random Forest for precise TTF prediction and Logistic Regression for practical early alerts to reduce downtime.

Keywords: Machine Learning · Predict Maintenance · Failure Detection · K-means Clustering · Logistic Regression · Random Forest · SMOTE

1 Introduction

1.1 Motivation

Machine failures can cause production interruptions, equipment damage, and higher repair costs if issues are not detected early. They can even impact the supply chain. Many industrial companies still rely on scheduled maintenance, which may lead to either over-maintenance or insufficient maintenance. Therefore, failure prediction plays an important role in industrial maintenance. It helps prevent failures and reduces unplanned downtime for equipment, machines, and processes. Failure prediction techniques analyze historical and real-time data to identify system conditions, events, and operational patterns.

Machine learning (ML) has become a common failure prediction technique because it can use time-series data to train prediction models, evaluate their performance, and even deploy them in production environments. Due to continuous improvements in machine learning algorithms, the growth of big data analytics and streaming platforms, and the increasing availability of industrial data, machine learning is now widely used for failure prediction. Many studies have shown that ML-based failure prediction models work well for different systems and areas, including agricultural machinery, aircraft components, and concrete properties, as well as manufacturing plants. Moreover, research indicates that machine learning is effective for both short-term predictions (such as failures occurring within minutes) and long-term predictions (such as failures occurring weeks later)[4][5].

For this study, machine failure dataset has been chosen as the foundation of our research. By analyzing machine operation data and applying machine learning algrorithms, this study aims to gain deeper insights into the factors that contribute to machine failures, and develop predictive models to anticipate potential breakdowns. In this way, the factory can enhance operational efficiency, and reduce unexpected downtimes, ultimately leading to cost savings and improved reliability.

1.2 Problem Description

Unplanned machine failures can lead to operational disruptions, increased maintenance costs, and even reduced industrial efficiency. However, traditional scheduled maintenance strategies often result in unnecessary servicing or missed failures, thus leading to an increase in maintenance cost and operation cost. Therefore, if the machine failure can be predicted, the factory can not only improve the operation efficiency, but also reduce unnecessary costs, contributing to enhancement of overall productivity.

1.3 Research Questions

This study aims to explore machine failure prediction by addressing the following three key questions. First, what are the different failure modes of machines and what operational characteristics define each mode. Second, can we utilize machine operational data to differentiate between normal and abnormal machine state. Finally, can we accurately predict the occurrence and timing of machine failures based on real-time feature states. By answering these questions, this research seeks to enhance operational efficiency and reduce unnecessary costs in industrial settings.

1.4 Literature Review

Failure detection and diagnosis have been widely studied in recent years due to their critical role in improving industrial operations. Early detection of failures helps minimize downtime, reduce maintenance costs, and prevent safety hazards. By predicting failures in advance, businesses can implement timely maintenance, ensuring that production processes remain efficient and uninterrupted. This proactive approach not only enhances the reliability of equipment but also extends its lifespan. As industries continue to adopt more complex systems, the importance of accurate and early failure detection becomes even more pronounced, making it a central focus for research in predictive maintenance.

In the field of machine learning, classification tasks aim to assign categories or labels to data points. Chawla et al. [3] studied the ideal scenario for datasets, suggesting that each category should have roughly the same number of samples to enable algorithms to learn and generalize effectively based on a balanced distribution. However, in real-world applications, datasets are often imbalanced, with some categories having significantly fewer instances than others. This imbalance is prevalent in various domains such as fraud detection, medical diagnosis, spam filtering, and hardware failure detection.Thai-Nghe et al. [9] investigated the behavior of traditional machine learning algorithms on imbalanced data, finding that these algorithms tend to favor the majority class since they typically optimize for overall accuracy. For instance, in a dataset with a low disease prevalence, a model predicting all instances as "not sick" might achieve high accuracy but fail to identify the few critical cases. Chawla et al. [3] further explored evaluation methods, arguing that relying solely on accuracy is insufficient for assessing model performance on imbalanced datasets, and they recommended more sensitive metrics such as precision, recall, and the F1 score, particularly when identifying the minority class is critical.

The common ways to deal with imbalanced data are under- and over-sampling. Undersampling techniques aim to balance class distributions by reducing the number of majority class instances. However, this approach can lead to the loss of valuable information, potentially discarding important patterns within the data. Oversampling methods, such as the Synthetic Minority Over-sampling Technique **(SMOTE)**, generate synthetic instances for the minority class by interpolating between existing samples and their neighbors. Blagus and Lusa [2] studied oversampling techniques to balance the class distribution by increasing the number of minority class samples. They found that random oversampling, the simplest approach, duplicates existing minority class samples. While it helps balance the dataset, it often leads to overfitting, causing the model to perform well on training data but poorly on unseen data. To address this issue, Chawla et al. [3] developed SMOTE, which creates synthetic samples by interpolating between minority class instances, improving data diversity and mitigating overfitting more effectively.

The algorithms most frequently adopted in machine failure detections and predictions were RF, SVM, and ANN. A few studies also adopted decision tree, gradient boosting k-nearest neighbor, and Naive Bayes[5].**Random Forest (RF)** is an ensemble learning algorithm that constructs multiple decision trees and aggregates their outputs to enhance classification accuracy and mitigate overfitting. Its robustness to noise, capacity to handle high-dimensional data, and ability to assess feature importance make it a preferred method for machine failure prediction. In a study by Shen et al. (2018)[8], RF was employed to predict hard disk drive failures using SMART attributes, achieving high precision and a low false alarm rate. The model used a part-voting strategy to improve prediction performance across different drive families. Similarly, Quiroz et al. (2017)[7] applied RF to detect broken rotor bar faults in line-start permanent magnet synchronous motors, extracting time domain features from transient current signals

and attaining an accuracy of 98.8%. These studies demonstrate RF's effectiveness in various industrial contexts, highlighting its adaptability to different types of machinery and failure modes.
Gradient Boosting (GB) is another powerful ensemble learning method that builds a series of weak learners, typically decision trees, in a sequential manner. Each new tree corrects the errors made by the previous one, making the model highly effective in capturing complex patterns within the data. GB is particularly known for its high predictive accuracy and ability to handle both regression and classification tasks. Unlike Random Forest, which constructs independent trees, GB focuses on reducing residual errors through its iterative process. Zhang et al. (2021) [10]implemented XGBoost for bearing fault prediction in rotating machinery, achieving 98.2% accuracy by optimizing hyperparameters to capture early vibration anomalies. In CNC machine monitoring, Chen et al. (2020)[6] developed a LightGBM framework that reduced false alarms by 40% compared to traditional SVM approaches through sequential error correction of thermal sensor patterns. Comparative analyses by Wang et al. (2022)[1] revealed GB's superior performance over random forest in dynamic industrial environments, particularly when handling imbalanced failure datasets through weighted loss functions. These advancements highlight GB's evolving role in predictive maintenance systems for modern smart manufacturing.

2 Data set preparation and data analysis

2.1 Data Description

This dataset comprises 149,855 records with 16 variables, describing the operational status and failure events of 205 machines. The following analysis details the distribution and statistical characteristics of each variable.

Basic Data Information The dataset has no missing values, with all variables fully populated. It includes 4 integer columns (`ID`, `WELL_GROUP`, `EQUIPMENT_FAILURE`, `AGE_OF_EQUIPMENT`), 8 float columns (`S15`, `S17`, `S13`, `S5`, `S16`, `S19`, `S18`, `S8`), and 4 object columns (`DATE`, `REGION_CLUSTER`, `MAINTENANCE_VENDOR`, `MANUFACTURER`).

Number of Machines The `ID` field represents a unique identifier for each machine. The dataset includes 205 unique machines, indicating that the analysis covers 205 distinct devices.

Categorical Variable Distribution The categorical variables include `REGION_CLUSTER` (region of the machine), `MAINTENANCE_VENDOR` (maintenance vendor), `MANUFACTURER` (manufacturer), and `WELL_GROUP` (machine type). As shown in Figure 1, the distributions are as follows:

– **Region of Machine**: Machines are distributed across 8 regions, with Region A (31,433 machines) and Region B (29,971 machines) being the most common, while Region F (4,386 machines) is the least represented, indicating an uneven regional distribution.
– **Maintenance Vendor**: There are 8 vendors, with Vendor M (24,123 machines) and Vendor O (21,930 machines) servicing the most machines, and Vendor L (10,234 machines) the least, showing a relatively balanced distribution.
– **Manufacturer**: The dataset includes 10 manufacturers, with T and S each having 19,737 machines, and R having the fewest at 5,848, reflecting slight variations in manufacturer representation.
– **Machine Type**: Machines are categorized into 4 types, with Type 2 (40,936 machines) and Type 3 (38,012 machines) being the most common, and Type 1 (35,088 machines) the least, showing a fairly even distribution.

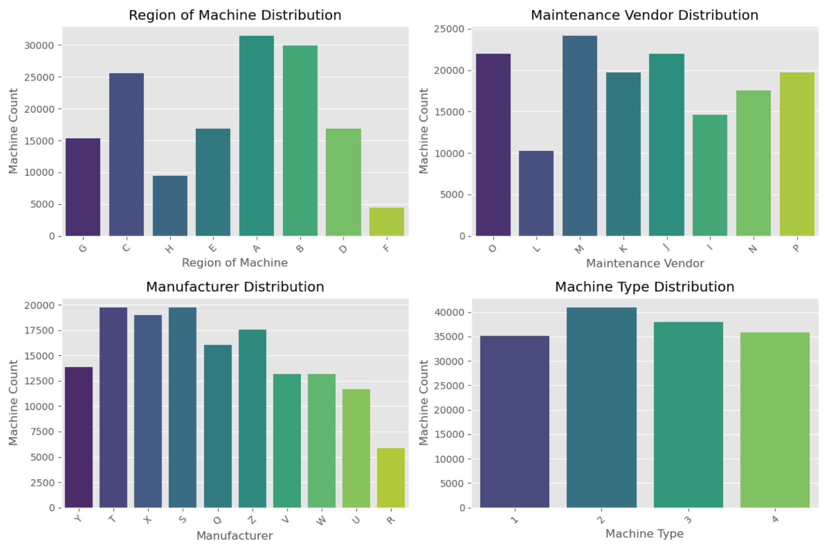

Fig. 1. Distribution of Categorical Variables: Region, Maintenance Vendor, Manufacturer, and Machine Type.

Sensor Data and Equipment Age The numerical variables include sensor values (`S15`, `S17`, `S13`, `S5`, `S16`, `S19`, `S18`, `S8`) and equipment age (`AGE_OF_EQUIPMENT`). Their statistical characteristics and distributions are shown in Figure 2:

– **Sensor Value S15**: Mean is 14.19, standard deviation 8.67, ranging from 0 to 51.23, with a right-skewed distribution concentrated between 0 and 30.

- **Sensor Value S17**: Mean is 85.92, standard deviation 85.57, ranging from 0 to 478.47, right-skewed, with most values between 0 and 200.
- **Sensor Value S13**: Mean is 35.64, standard deviation 14.72, ranging from 0 to 570.51, approximately normally distributed, concentrated between 20 and 50.
- **Sensor Value S5**: Mean is 4,557.52, standard deviation 2,497.99, ranging from 0 to 52,767, right-skewed, concentrated between 0 and 10,000.
- **Sensor Value S16**: Mean is 7.93, standard deviation 2.43, ranging from 0 to 23.53, approximately normally distributed, concentrated between 5 and 15.
- **Sensor Value S19**: Mean is 8.23, standard deviation 13.04, ranging from 0 to 291.10, right-skewed, concentrated between 0 and 20.
- **Sensor Value S18**: Mean is 110.08, standard deviation 197.36, ranging from 0 to 3,995.90, extremely right-skewed, with most values between 0 and 500.
- **Sensor Value S8**: Mean is 117.48, standard deviation 204.11, ranging from -16.49 to 1,824.96, right-skewed, concentrated between 0 and 500, with a few negative values (potentially anomalies).
- **Equipment Age**: Mean is 2,751.15 days (approximately 7.5 years), standard deviation 3,368.35, ranging from 0 to 15,170 days (approximately 41.5 years), right-skewed, with most ages between 0 and 5,000 days.

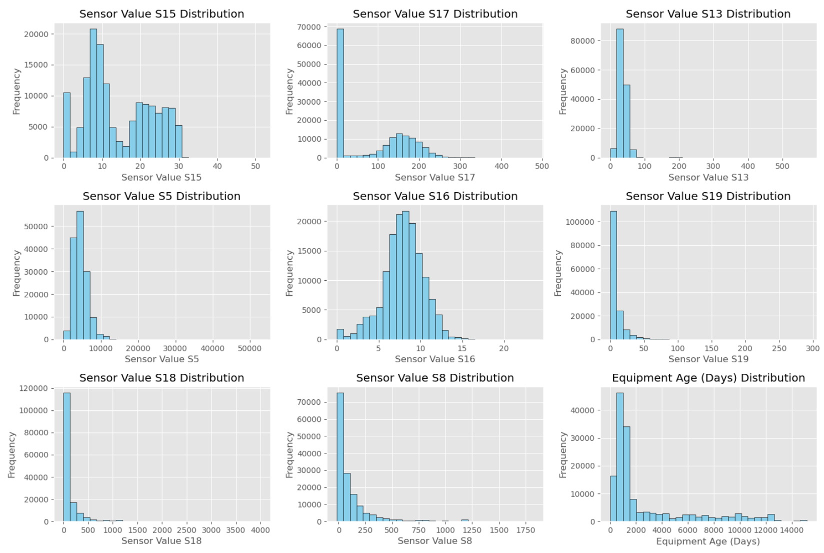

Fig. 2. Distribution of Numerical Variables: Sensor Values and Equipment Age.

Equipment Failure Distribution The `EQUIPMENT_FAILURE` variable indicates failure status (0 for no failure, 1 for failure). As shown in Figure 3, the distribution is highly imbalanced: no failure (0) accounts for 99.86% of observations, while failure (1) accounts for only 0.14%, indicating that failure events are extremely rare.

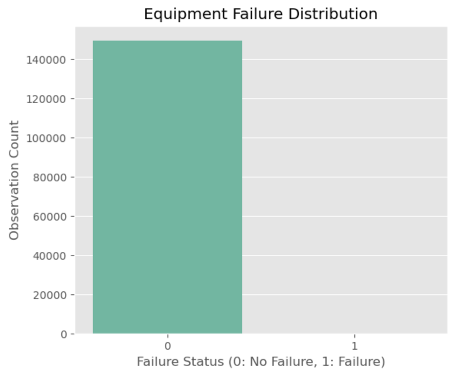

Fig. 3. Distribution of Equipment Failure.

2.2 Imbalance Problem

The extreme imbalance in the `EQUIPMENT_FAILURE` variable poses significant challenges for predictive modeling. This imbalance means that out of 149,855 observations, only approximately 210 records correspond to failure events, while the remaining 149,645 are non-failure events. Such a unbalanced distribution can lead to models that are biased toward predicting the majority class (no failure), potentially overlooking the rare but critical failure events. This issue is visually evident in Figure 3, where the bar for failures (1) is barely visible compared to the dominant non-failure bar (0). To address this in subsequent analyses, techniques such as oversampling the minority class (e.g., using SMOTE), undersampling the majority class, or applying class-weighted loss functions during model training may be necessary to improve the model's ability to detect failures.

2.3 Data Preprocessing

SMOTE method In this study, SMOTE is applied to an equipment failure dataset with 149856 records, where only 206 instances represent failures. The dataset includes eight numerical features (`S15`, `S17`, `S13`, `S5`, `S16`, `S19`, `S18`, `AGE_OF_EQUIPMENT`) and several categorical features. To balance the classes, SMOTE is used on the numerical features only, targeting 3,000 minority samples. The steps are: (1) extract numerical features, (2) apply SMOTE to generate 2,796 additional failure instances, and (3) merge with original data. The result is a dataset of 152,650 records (149,650 non-failures and 3,000 failures), retaining just the numerical features and the target variable for machine learning tasks.

Calculation of Time To Failure The goal of this step is to calculate the `Time to Failure` (TTF) for each device (`ID`), retaining records from the start to the last failure and discarding data after the last failure so that the dataset got the TTF column data which is used as input data in section 5

The input is sourced from original dataset, with key columns `ID`, `DATE`, `EQUIPMENT_FAILURE` (0 for no failure, 1 for failure), and `AGE_OF_EQUIPMENT`, spanning approximately 150,000 rows.

The process begins with loading the CSV into a DataFrame and sorting it by `ID` and `AGE_OF_EQUIPMENT` to establish a chronological order. Next, filtering retains only `ID`s with at least one `EQUIPMENT_FAILURE == 1`, excluding devices without failures. TTF calculation follows: the `Next_Failure_Age` is derived using `groupby('ID')` and `shift(-1)` to identify the `AGE_OF_EQUIPMENT` of the next failure, then TTF is computed as `Next_Failure_Age - AGE_OF_EQUIPMENT`, with failure rows set to 0. Rows after the last failure are removed by comparing `AGE_OF_EQUIPMENT` to the maximum failure age per `ID`. Finally, the result is saved to a new CSV file.

3 Identifying Operational Characteristics of Failure Modes via Clustering

To address the first research question—investigating the different failure modes of machines and identifying the operational characteristics that define each mode—this section applies unsupervised clustering techniques to failure-related data. This analysis helps differentiate between various operational signatures of failure events, laying a foundation for further predictive modeling. The clustering results are interpreted both numerically and categorically to provide comprehensive insights into machine behavior during failure.

3.1 Rationale and Implementation (Why and How)

To explore potential failure modes in equipment operation, we employed a data-driven approach using K-means clustering. Given the variety of sensor readings and categorical information in the dataset, this unsupervised method allows us to group failure cases based on similarity in sensor patterns, potentially revealing different failure conditions or equipment states. K-means was chosen for its simplicity and effectiveness in handling high-dimensional numerical data. In the following subsections, we describe the data preparation process, the clustering procedure, and the resulting insights.

3.2 Preprocessing and Clustering Procedure

The analysis focused exclusively on equipment failure records. Sensor data columns were combined with categorical variables such as region cluster and maintenance vendor. To enable clustering, categorical features were encoded using label encoding, and all features were standardized to eliminate bias from differing feature scales.

To determine the optimal number of clusters, the Elbow Method was applied by plotting the sum of squared errors (SSE) against increasing values of k. As shown in Figure 4, the inflection point occurs at $k = 7$, indicating diminishing returns in reducing SSE beyond that point. Thus, seven clusters were selected to balance complexity and explanatory value.

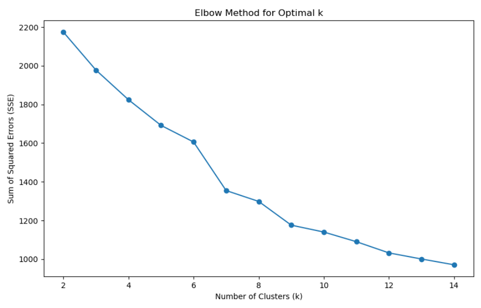

Fig. 4. Elbow Method for Optimal k

3.3 Results Analysis of Clustering

Once the clusters were established, their characteristics were visualized using a radar chart (Figure 5). This chart highlighted distinct sensor patterns across clusters. For example, Cluster 6 exhibited significantly elevated values in sensors S15 and S18 compared to other clusters, while Cluster 0 showed marked deviations in sensors S5 and S8. Cluster 2 displayed unusually low readings in S13 and S18, and Cluster 3 had moderate but consistent anomalies across multiple sensors. These patterns suggest that different clusters may represent unique combinations of sensor values, though the exact physical interpretation of these sensors requires domain-specific knowledge.

For categorical variables, the standardized numerical values in the radar chart provided limited interpretability. To address this, the data for each cluster was exported to an Excel file, where categorical variables were analyzed separately. To analyze the distribution of categorical variables across clusters, statistical summaries and visualizations. These charts (Figures 6–9) explicitly show the proportion of each category (e.g., vendors labeled I, J, K) within individual clusters.

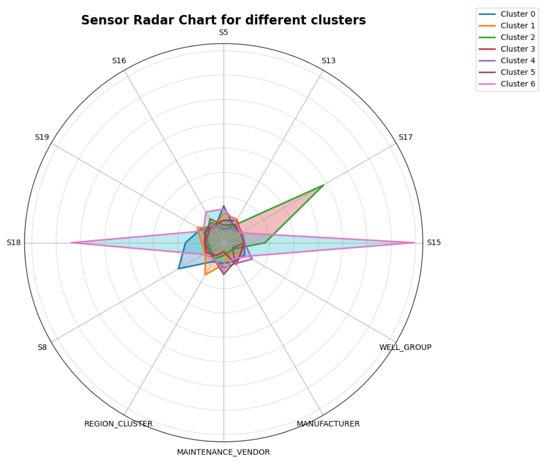

Fig. 5. Radar Chart of Cluster

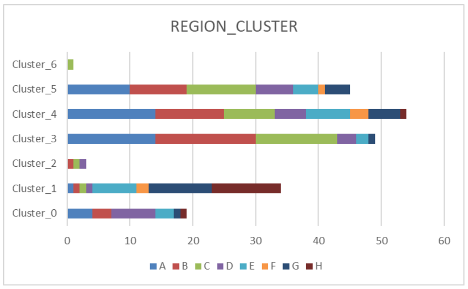

Fig. 6. Distribution of Region Cluster Categories within Individual Clusters

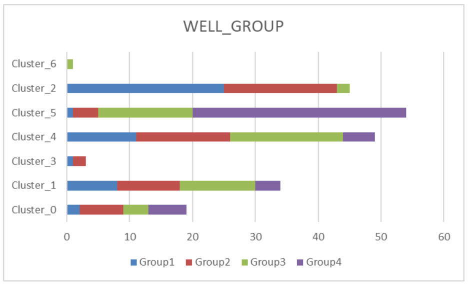

Fig. 9. Distribution of Well Group within Individual Clusters

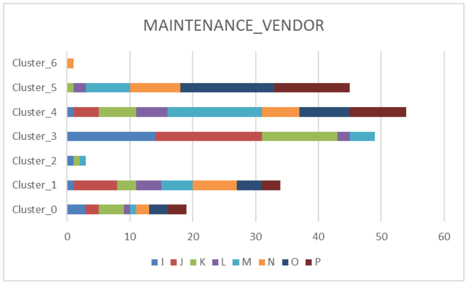

Fig. 7. Distribution of Maintenance Vendor Categories within Individual Clusters

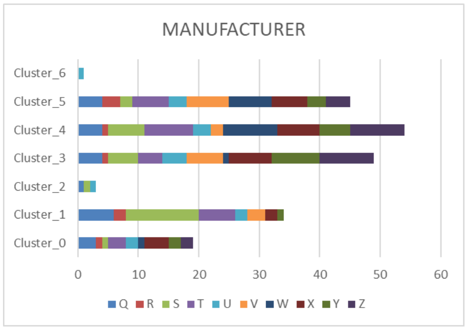

Fig. 8. Distribution of Manufacturer Categories within Individual Clusters

Under region cluster, Cluster 0 is strongly associated with region "D", while Cluster 3 shows a concentration in region "A" and “B”. For maintenance vendor, Cluster 3 is dominated by vendor "J" while cluster 5 is equally dominated by vendor "O". In manufacturer, Cluster 1 stands out with manufacturer "S", suggesting potential equipment-specific failures from this supplier. The well group variable reveals that both Cluster 5 is heavily skewed toward "Group4" (over 50%), indicating operational or environmental factors specific to this group. These findings emphasize that while sensor data drives primary cluster differentiation, categorical variables like vendor, region, and well group provide critical supplementary insights for targeted root-cause analysis.

4 Determining Equipment State Based on Operational Data

To address the second research question—whether machine operational data can be used to differentiate between normal and abnormal machine states—this section investigates the feasibility of identifying equipment state (failure or non-failure) using known operational data. We presents two machine learning methods applied to determine the equipment state: linear regression and logistic regression.We utilize a dataset that includes eight numerical features (`S15`, `S17`, `S13`, `S5`, `S16`, `S19`, `S18`, `AGE_OF_EQUIPMENT`), generated using the SMOTE algorithm. All of the methods are trained using the same dataset and feature set, which includes eight numerical sensor readings and equipment age. For each method, we validate the performance using metrics on a separate test dataset. We then analyze the results to assess model effectiveness and to identify which features—particularly sensor variables—contribute most to predicting equipment failure. Linear regression is used as a baseline for evaluating the performance of logistic regression.

4.1 Performance Metrics Summary

In this study, we employ several performance metrics to evaluate the performance of our models. These metrics include:

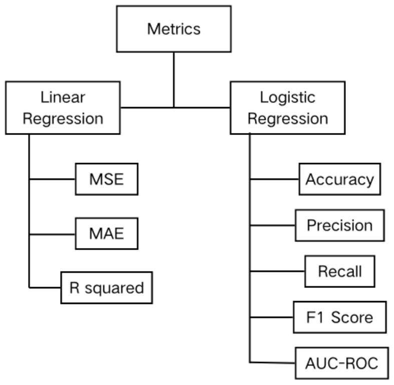

Fig. 10. Performance Metrics Summary

- **Accuracy**: The ratio of correctly predicted observations to the total observations.
- **Precision**: The ratio of true positive predictions to the total predicted positives.
- **Recall**: The ratio of true positive predictions to the actual positives.
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two.
- **AUC-ROC**: The area under the Receiver Operating Characteristic curve, which evaluates the model's ability to distinguish between classes.
- **Mean Squared Error (MSE)**: The average of the squared differences between predicted and actual values.
- **Mean Absolute Error (MAE)**: The average of the absolute differences between predicted and actual values.

– **R-squared (R^2 Score)**: Represents the proportion of variance in the dependent variable that is predictable from the independent variables.

The formulas for these metrics are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$
$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{AUC-ROC} = \int_0^1 \text{TPR}(t)\, d\text{FPR}(t)$$
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where:

– TP – True Positives
– TN – True Negatives
– FP – False Positives
– FN – False Negatives
– $\text{TPR}(t)$ – True Positive Rate as a function of threshold t
– $\text{FPR}(t)$ – False Positive Rate as a function of threshold t
– y_i – Actual value
– $\hat{y}_i$ – Predicted value
– $\bar{y}$ – Mean of actual values
– n – Number of observations

4.2 Linear Regression

Model Rationale and Implementation (Why and How) The choice of linear regression is driven by three key considerations: First, its simplicity and computational efficiency enable rapid validation of potential linear relationships between features (sensor values) and the target variable. Second, the model's strong interpretability allows straightforward analysis of critical feature contributions to failure prediction, establishing an interpretable benchmark for subsequent complex models. Third, as a baseline model, it provides a clear reference for quantifying performance improvements in advanced models (Logistic Regression, Random Forest, Gradient Boosting).

Model Introduction Linear regression is a fundamental statistical method used to model the relationship between a dependent variable and one or more independent variables.The core function is a linear equation, which represents the dependent variable as a weighted sum of the independent variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$$

Where:

- Y is the dependent variable (the predicted outcome)
- β_0 is the intercept
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for the independent variables
- $X_1, X_2, \ldots, X_n$ are the independent variables (features)
- ε is the error term, representing the difference between the observed and predicted values

The loss function commonly used for linear regression is the Mean Squared Error (MSE), which measures the average squared difference between the actual values and the predicted values:

$$Loss = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

where:

- N is the number of samples
- y_i is the actual value of the dependent variable for sample i (the true label (0 or 1))
- $\hat{y}_i$ is the predicted lable for sample i

To find the optimal parameters β, we minimize the MSE loss function using the Least Squares Method, which provides a closed-form solution:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

where:

- $\mathbf{X}$ is the design matrix containing all feature values
- $\mathbf{y}$ is the vector of actual target values
- $\boldsymbol{\beta}$ is the vector of coefficients to be estimated
- $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ is the pseudoinverse of $\mathbf{X}$, which projects $\mathbf{y}$ onto the feature space

Alternatively, for large datasets where computing the matrix inverse is expensive, Gradient Descent can be used to iteratively optimize β:

$$\beta_j \leftarrow \beta_j - \alpha \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i) x_{ij}$$

where:

- α is the learning rate
- The term $(y_i - \hat{y}_i)x_{ij}$ represents the gradient of the loss function with respect to β_j

As the loss function converges to a minimum, the parameters β are finalized. The trained model can then be used to make predictions on new data.

Validation and Result Analysis The evaluation metrics reveal key limitations in using linear regression for machine failure prediction. The low R^2 score (0.0143) indicates that the model explains only 1.4% of the variance in equipment failure, suggesting weak alignment between features and the target variable. While the MSE (0.0015) and MAE (0.0043) appear numerically small, these values reflect the binary nature of the target variable (0/1), where minor absolute deviations between predictions and true values inflate performance optimism. For instance, predicting near-zero probabilities for most non-failure cases artificially reduces error metrics but fails to capture meaningful failure patterns. This mismatch highlights the fundamental unsuitability of linear regression for classification tasks, as its unbounded outputs lack probabilistic interpretation. To improve predictive validity, transitioning to logistic regression or tree-based models would better align with the binary outcome structure while enabling calibrated probability estimates.

Evaluation Feature importance analysis reveals extremely weak coefficients (average absolute impact: 0.0015), with most sensors showing negligible negative correlations, implying no meaningful linear relationships. As a baseline, these results highlight fundamental limitations in the data's linear signal, urging exploration of nonlinear methods or improved feature engineering. The model's failure establishes a critical reference point for evaluating future enhancements in feature selection or algorithmic complexity.

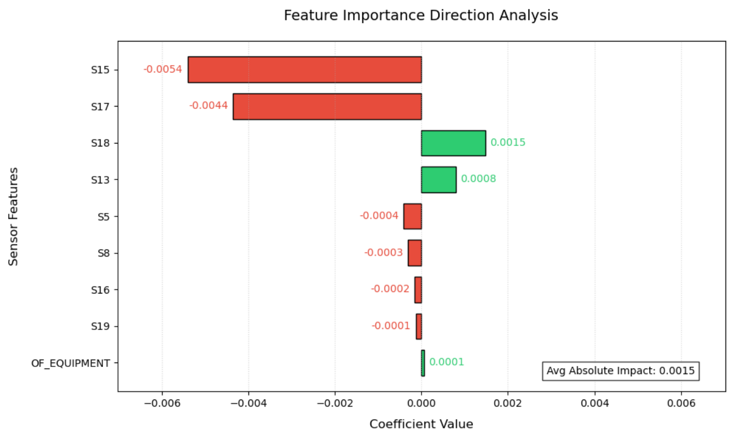

Fig. 11. Feature Importance of Coefficients

4.3 Logistic Regression

Model Rationale and Implementation (Why and How) Logistic regression is a widely used and interpretable method for binary classification, making it a natural choice for predicting equipment failure based on operational data. Unlike linear regression, which assumes a continuous outcome, logistic regression models the probability of a binary state—failure or normal—based on input features. Another advantage of logistic regression is that its coefficients directly reflect the relationship between each feature and the likelihood of failure, helping identify key contributing factors in sensor data. This interpretability is valuable for drawing insights beyond raw prediction.

Given the alignment of logistic regression with the classification nature of the task, it further supports the rationale for using linear regression as a baseline for comparison.

The logistic regression model was implemented using the same feature set as the baseline, including eight numerical sensor readings and equipment age. SMOTE was applied to the training data to address class imbalance, and model training was performed by minimizing binary cross-entropy loss. All performance metrics were computed using predictions on a separate test dataset.

Model Introduction Logistic regression predicts the probability of a binary outcome using the logistic (sigmoid) function, which transforms a linear combination of input features into a probability between 0 and 1:

$$P(y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n)}}$$

where $P(y = 1)$ is the probability of the positive class (equipment failure), β_0 is the intercept, $\beta_1, \ldots, \beta_n$ are feature coefficients, and $x_1, \ldots, x_n$ are input values. The model is trained by minimizing the binary cross-entropy loss:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right]$$

where N is the number of samples, y_i is the true label (0 or 1), and p_i is the predicted probability.

Validation and Result Analysis The model is evaluated on a held-out test set using classification metrics. Table 1 summarizes the performance of logistic regression.

Table 1. Logistic Regression Performance on Test Set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Non-failure (0)	0.99	0.63	0.77	—
Failure (1)	0.04	0.71	0.08	—
Accuracy		0.6342		
AUC-ROC		0.7504		

When evaluated on the test set, the classification report highlights the impact of class imbalance. The model achieves high precision (0.99) but relatively low recall (0.63) for the non-failure class, indicating it successfully avoids false positives but misses some actual non-failures. For the failure class, recall is high (0.71), suggesting strong sensitivity to failure cases, but precision drops significantly (0.04), meaning many of the predicted failures are actually false alarms.

This behavior is typical in imbalanced datasets and suggests that while the model is sensitive to failures, it lacks specificity. Although SMOTE was applied to augment the training data, the minority class remains underrepresented overall.

Analyzing the feature importance of coefficients (Figure 12), S18 emerges as the most positive predictor of failure and could be a key factor contributing to equipment failure. This sensor should be closely monitored in production operations. The feature values of AGE_OF_EQUIPMENT, S19, and S16 have very small absolute values, and their impact on equipment status can be ignored in this model.

Evaluation Compared to the linear regression baseline, logistic regression shows a significantly better ability to detect failures. While linear regression achieved higher accuracy and AUC, it failed to identify any positive cases, resulting in zero precision and recall for the failure class. Logistic regression, though less precise, offers much better recall and classification balance, making it a more effective choice for failure prediction in this context.

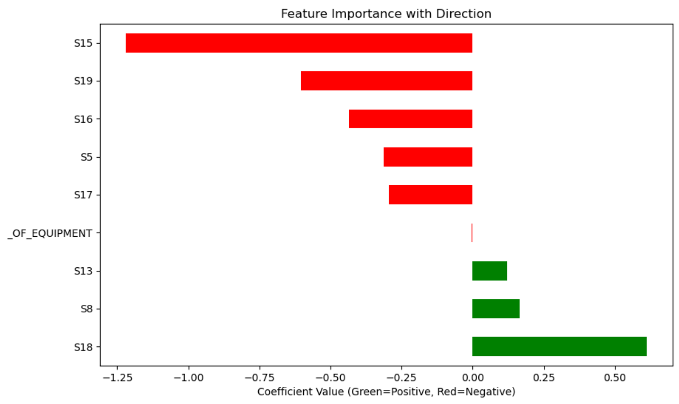

Fig. 12. Feature Importance of Coefficients

5 Prediction of Machine Failure Occurrence

In this section, the research question of whether machine learning models can accurately predict the occurrence of machine failures is studied. The timing of machine failure is predicted based on real-time sensor states. Several machine learning methods are applied for prediction, with a linear model used as a benchmark to compare the accuracies of different models.

5.1 Evaluation Metrics

Several evaluation metrics are used to assess the performance of the regression model. All of them have been introduced in the abovementioned paragraph, including mean squared error (MSE) and R-Squared (R^2), accuracy, precision, recall, F1-score.

5.2 Linear Regression

Basic Linear Regression The first method applies a standard linear regression model, where the eight sensor values are used as independent variables. Before training, all features were standardized to ensure consistent scaling across variables. The dataset was split into training and testing sets, with 20% reserved for testing. The results indicate limited predictive performance. An R^2 value is only 0.0270 and a MSE is 31,075.98 for this model.

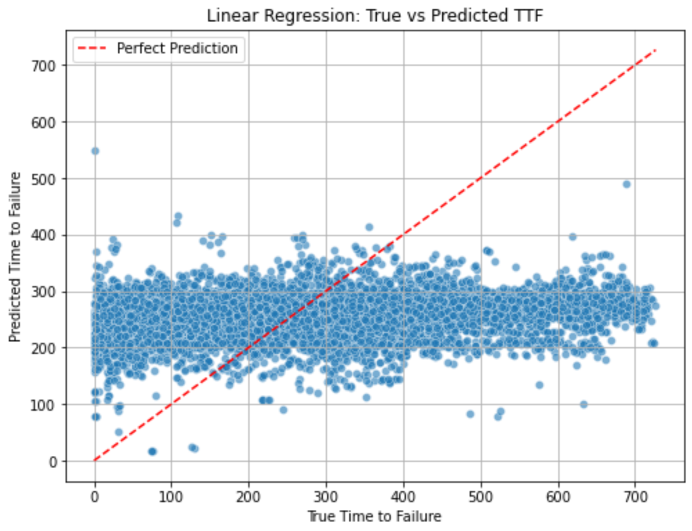

Fig. 13. Linear Regression for Timing to Failure

Linear Regression with Non-linear Feature Transformations In the second model, non-linear transformations were applied to selected sensor values to improve performance, as the basic model showed limited results. The squares of S13 and S8 were added to capture potential quadratic effects. A logarithmic transformation was applied to S5 to reduce the influence of extreme values, since S5 has a wide value range. Additionally, an interaction term between S16 and S18 was included to explore whether their combined effect is related to equipment failure. After applying these transformations, the model showed a slight improvement with an R^2 score of 0.0398 and a MSE of 30,667.10.

Linear Regression with One-Hot Encoding The third method builds upon the previous models by introducing one-hot encoding for categorical variables. The categorical features region cluster, maintenance vendor, manufacturer, and well group were encoded using one-hot encoding. This allows the model to consider categorical information that may be relevant to equipment failure. The model performance further improved but still not really significantly. A MSE is 30,284.54 and an R^2 score is 0.0518 for this model.

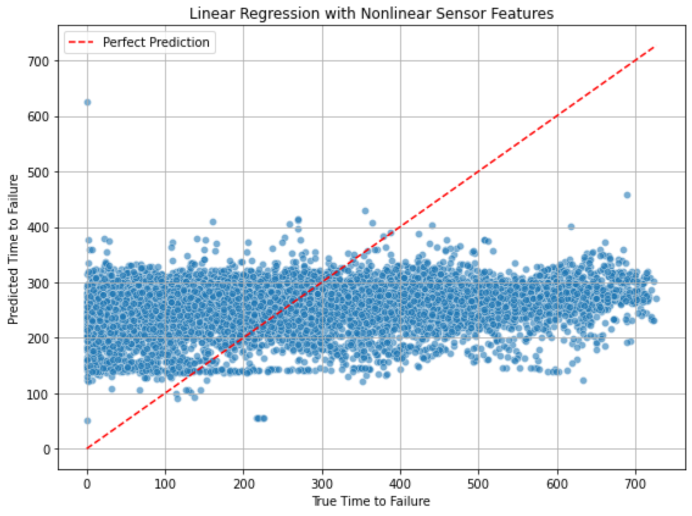

Fig. 14. Linear Regression with Non-linear Feature Transformations

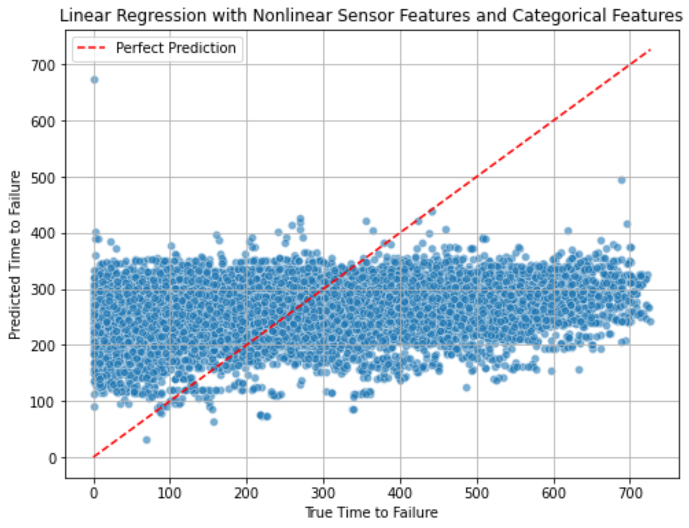

Fig. 15. Linear Regression with One-Hot Encoding

5.3 Multiple Machine Learning Methods

In this section, multiple machine learning methods are implemented to predict the timing to machine failure. Logistics Regression model, Random Forest model, and Gradient Boosting model are trained using the same training set and testing set involving the status of all 8 sensors. Another prediction accuracy indicator is added, which is whether the predicted time-to-failure result is within the range of 30 days in advance of the actual failure date. This indicator simulates the condition that the prediction result can help establish an alarm mechanism a month in advance. Prediction results of different methods are then validated, compared and discussed.

Logistics Regression The logistics regression model is trained based on the eight numeric sensor readings to predict the value of time to failure. Training set and testing set proportion is divided as 80%-20%. The result is evaluated using the same evaluation metric in 3.3. Evaluation result is presented in Table 2.

Table 2. Logistic Regression Prediction Performance on Test Set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Non-failure (0)	0.92	1.00	0.96	15567
Failure (1)	0.00	0.00	0.00	1301
Accuracy			0.9229	
AUC-ROC			0.6540	

As is shown in the table, the trained logistics regression model achieves high precision (0.92), recall (1.00), and F1-score (0.96) for the non-failure class, but disappointing accuracy (all zero) for the failure class. The overall accuracy is high (0.92) as non-failure class constitutes the majority of test set. However, the zeros in the failure class may not indicate the disappointing performance of logistics regression model in predicting time to failure. After all, the poor performance in failure class does not match the excellent prediction performance in non-failure class. Possible reasons may indicate the condition that the prediction is not strictly accurate, but is actually close to the actual machine failure time.

Therefore, the additional indicator of whether the predicted time-to-failure result is within the range of 30 days in advance of the actual failure date is introduced. All predictions within 30 days before machine failure time are viewed as successful predictions. Based on this newly added standard, the relaxed accuracy goes up to the incredible 1.00. This result indicates that although logistics regression model may struggle to give the precise correct date of machine failure, it can help predict and alarm potential machine failure in one month with a high accuracy.

Feature importance of coefficients is presented in Figure 16. S17, S15, and S19 are identified as the three most contributing negative influencers, while S18 and S8 has a weaker positive impact. Other sensors are not calculated as major influencing factors.

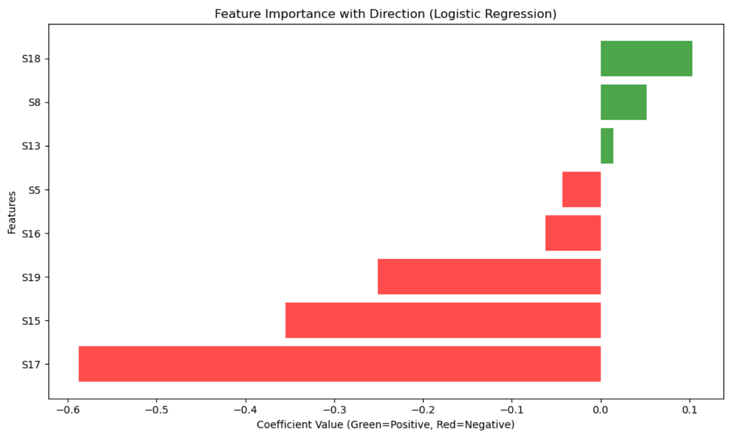

Fig. 16. Feature Importance of Logistics Regression Prediction

Random Forest In addition to the logistic regression model, other machine learning approaches are explored to evaluate potential improvements in prediction performance. Among these, the random forest model is implemented due to its well-known strength in achieving high prediction accuracy.

The random forest method is an ensemble learning technique that builds multiple decision trees during training. Each tree is constructed using a different subset of the training data, selected through random sampling with replacement. At each split in the tree, a random subset of features is considered, which helps to reduce correlation between trees. During prediction, the outputs from all individual trees are aggregated—typically by majority vote for classification tasks or by averaging for regression tasks. This process helps to reduce overfitting and improves the model's generalization ability. As a result, random forests often outperform single decision trees, especially when dealing with complex or noisy datasets.

The model is trained and tested using the same training and testing set as the former logistics regression model. The model prediction performance is presented in Figure 17.

Compared to linear model, random forest shows great improvement in prediction accuracy. In accordance with the direct judgment from the graph, model performance evaluation metric gives the similar outcome. The MSE of the model is 4519.2, and the R^2 value is 0.8585. Both values show a giant increase in prediction accuracy. The relaxed accuracy indicator regarding prediction within a month's range is also calculated. The result is 0.6377, which is acceptable, but not as satisfactory as the result of logistics regression model.

Feature importance of sensors is also calculated, shown in Figure 18.

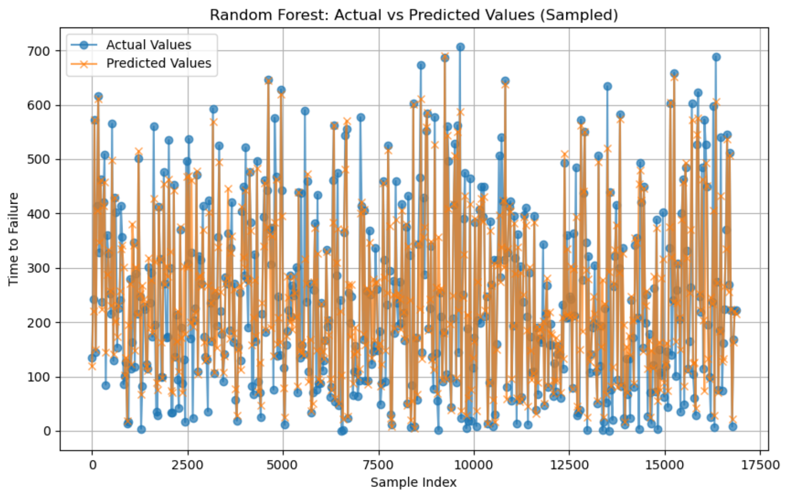

Fig. 17. Random Forest Model Prediction Performance

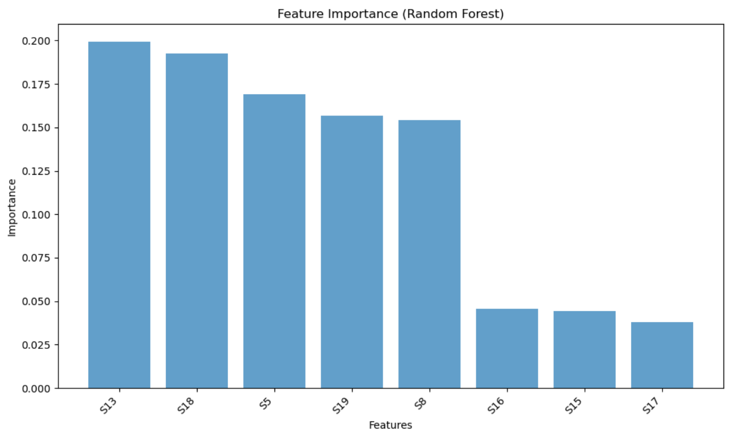

Fig. 18. Random Forest Feature Importance

As is shown in Figure 18, S13, S18, S5, S19, and S8 are the five leading contributing factors. S16, S15, and S17, on the other hand, play a slightly smaller role. The result shows some similar patterns with Figure 16, with S18, S13 on top and S15, S17 at the bottom. However, the two feature importance results don't give the same direct information after all. The diversity may also be the reason for the different prediction accuracy and relaxed accuracy. The variation in prediction mechanism can be the cause of this.

Gradient Boosting Besides random forest, gradient boosting is also implemented to further assess its potential in enhancing prediction performance. This method is widely recognized for its strong predictive power, especially in structured data tasks.

Gradient boosting is an ensemble technique that builds models in a sequential manner. Unlike random forest, which constructs trees independently, gradient boosting adds each new tree to correct the errors made by the previous ones. At each step, the model fits a new decision tree to the residual errors of the current prediction. The process continues for a set number of iterations or until the performance stops improving. By combining many weak learners—typically shallow decision trees—gradient boosting can form a strong overall model. Regularization techniques such as learning rate control and tree depth limitation are used to prevent overfitting and improve generalization.

The gradient boosting model is trained and tested still using the same training and testing dataset as the logistics regression model and random forest model. The gradient boosting model prediction performance is presented in Figure 19.

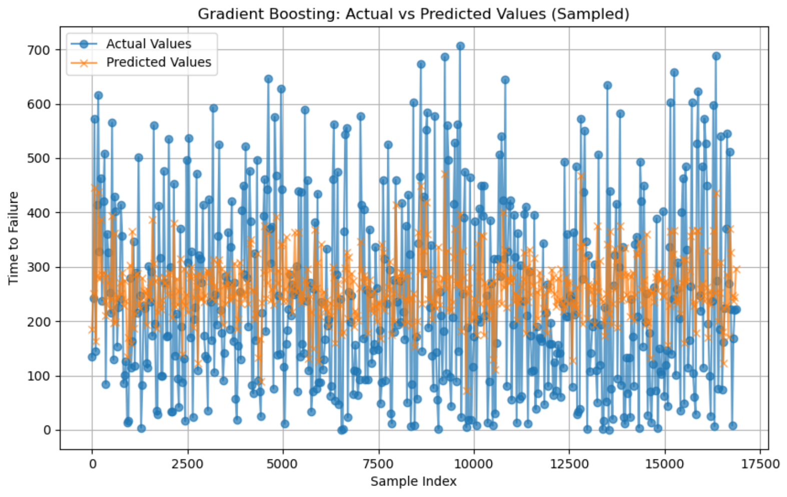

Fig. 19. Gradient Boosting Model Prediction Performance

The prediction result of gradient boosting model is more similar to the linear regression model, rather than the random forest model. Both gradient boosting and linear regression shows their struggle in interpreting the trained and tested data. Aligning with the graph, the prediction performance parameter of gradient boosting model is disappointing compared to random forest ones. The MSE is 24939.10, and the R^2 is 0.2191. To further demonstrate the poor prediction results, the relaxed accuracy is also calculated, with an unpleasant value of 0.1261, indicating only 12.61% samples are predicted relatively accurately. Therefore, there is no need to further investigate the feature importance of gradient boosting model.

Result Discussion To sum up and assess all the used machine learning methods, related results are evaluated and discussed together. All the performance and above mentioned outcomes are integrated into the following table.

Model Selection	Linear Regression	Logistics Regression	Random Forest	Gradient Boosting
Model Performance	MSE: 30284.54	Accuracy: 0.9229	MSE: 4519.2	MSE: 24939.1
	R^2: 0.0518	AUC-ROC: 0.6540	R^2: 0.8585	R^2: 0.2191
Relaxed Accuracy	—	1	0.6377	0.1261

Table 3. Model performance comparison

As is shown in Table 3, the linear regression model and the three machine learning models achieve a distinct prediction result. Selected as comparison baseline, the linear regression model performs poorly in traditional evaluation metrics. The logistics regression model performs well in non-failure class, but poorly in failure class, achieving an acceptable but not satisfactory score in accuracy and AUC-ROC. The random forest model performs best in traditional evaluation metrics, with a MSE of 4519 and a R^2 of 0.8585. Only achieving a low score in MSE and R^2, the gradient boosting model is outperformed by the other two machine learning methods. Among the three machine learning models, random forest performs best in traditional model performance evaluation metrics, followed by an acceptable result of logistics regression and an unacceptable result of gradient boosting. Besides traditional evaluation metrics, the newly introduced relaxed accuracy indicating prediction correctness in 30 days is looked into. Regarding this evaluation metric, logistics regression achieves an incredibly trustworthy performance, while random forest achieves an acceptable accuracy, with gradient boosting lagged behind.

In summary, the logistics regression model and random forest model are the two suitable models for predicting machine failure events given multiple sensor readings. The random forest model is preferred when higher precision is required across both failure and non-failure classes, due to its balanced and superior performance in traditional metrics. It demonstrates strong predictive capability, especially with a high R^2 value, suggesting better generalization to unseen data. On the other hand, the logistics regression model, despite its relatively weaker performance in the failure class, exhibits strong performance under the relaxed accuracy metric. This indicates its potential for practical application scenarios where early warning within a flexible time window is acceptable.

6 Conclusion

This study aimed to understand and predict machine failures using machine learning methodologies, focusing on three primary research questions. The conclusions drawn are as follows:

Answers to Research Questions

1. **What are the different failure modes of machines and the operational characteristics that define each mode?**
   Through K-means clustering on failure records, the study identified seven distinct potential failure modes. These modes differed not only in numerical sensor readings (showing unique combinations, e.g., in S15, S18, S5, S8, S13) but also correlated significantly with categorical variables like machine region, maintenance vendor, manufacturer, and equipment type. This indicates that different failure modes indeed correspond to specific operational characteristics and contextual factors.
2. **Can machine operational data be used to differentiate between normal and abnormal (failure) states?**
   The study confirmed that operational data contains information to distinguish between normal and failure states. Using Logistic Regression (after applying SMOTE for imbalance), the model successfully identified a majority of actual failure instances (Recall=0.71 for the failure class), significantly outperforming a baseline Linear Regression model that failed to detect any failures. The results clearly show that operational data is sufficient for differentiating machine states, with sensor S18 identified as an important feature.
3. **Can machine learning models accurately predict the occurrence and timing of machine failures based on real-time feature states?**
   The study finds out that ML models can predict the occurrence and timing (Time To Failure - TTF) based on real-time features, but model accuracy and suitability vary. For predicting the exact TTF, Random Forest performed best according to standard metrics (MSE, R^2=0.8585). Conversely, Logistic Regression, while less accurate by standard metrics, showed excellent performance (100% relaxed accuracy) in providing an early warning within a flexible window (e.g., 30 days prior), making it practically valuable. Therefore, the choice of model depends on whether prediction precision or a timely warning window is prioritized.

In summary, by applying machine learning techniques (clustering, classification, regression), this research successfully explored machine failure modes, validated the possibility of distinguishing states using operational data, and evaluated the capability of different models for predicting failure times. This provides valuable insights and methodologies for implementing more effective predictive maintenance, improving operational efficiency, and reducing costs.

References

1. Bentéjac, C., Csörgő, A., Martínez-Muñoz, G.: A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review **53**(1), 49–73 (2020). `https://doi.org/10.1007/s10462-020-09896-5`
2. Blagus, R., Lusa, L.: Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics* ***14**, 106 (2013). `https://doi.org/10.1186/1471-2105-14-106`*
3. *Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: Synthetic minority over-sampling technique.* J. Artif. Intell. Res. (JAIR) **16**, 321–357 (2002). `https://doi.org/10.1613/jair.953`
4. Kolokas, N., Vafeiadis, T., Ioannidis, D., Tzovaras, D.: A generic fault prognostics algorithm for manufacturing industries using unsupervised machine learning classifiers. Simulation Modelling Practice and Theory **103**, 102109 (2020). `https://doi.org/https://doi.org/10.1016/j.simpat.2020.102109`
5. Leukel, J., González, J., Riekert, M.: Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. Journal of Manufacturing Systems **61**, 87–96 (2021). `https://doi.org/https://doi.org/10.1016/j.jmsy.2021.08.012`
6. Li, Y., Chen, L., Wang, Y., Liu, L.: A fault diagnosis method for key components of the cnc machine feed system based on the doubleensemble–lightgbm model. Machines **12**(5), 305 (2024). `https://doi.org/10.3390/machines12050305`
7. Quiroz, J.C., Mariun, N., Mehrjou, M.R., Izadi, M., Misron, N., Radzi, M.A.M.: Fault detection of broken rotor bar in ls-pmsm using random forests. arXiv preprint arXiv:1711.02510 (2017)
8. Shen, J., Wan, J., Lim, S.J., Yu, L.: Random-forest-based failure prediction for hard disk drives. International Journal of Distributed Sensor Networks **14**(10), 1–13 (2018). `https://doi.org/10.1177/1550147718806480`
9. Thai-Nghe, N., Gantner, Z., Schmidt-Thieme, L.: Cost-sensitive learning methods for imbalanced data. In: *The 2010 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2010)*
10. *Wei, L., Hu, X., Yin, S.: Optimized-xgboost early warning of wind turbine generator front bearing fault. Journal of System Simulation **33**(10), 2335–2343 (2021)*

Contribution

Teamwork contribution is stated as follows:

Yumeng Pan: Research question set up, Part of the literature review, Data description and preprocessing, Conclusion

Minghao Li: Research question set up, Clustering Analysis, Logistics Regression, Random Forest & Gradient Boosting Modeling for machine failure prediction

Yilin Shi: Research question set up, Clustering Analysis, Logistic regression for machine failure detection

YungTing Lu: Research question set up, Motivation, Liniear regression for machine failure prediction

Xinyu Yang: Research question setup, literature review, linear regression for machine failure detection

Use of AI Statement

We use ChatGPT to check our code and ask it to write the comment. We also use ChatGPT to change the content into latex format to write the report such as 'change this table or formula into latex format'.

Code

All codes of our research in section 2, 3, 4, 5 are available at: `https://github.com/LuizLi/ME44312-Machine-Learning.git`