# Predictive Maintenance on Industrial and Machine Sensor Data

Ahmad Al Jaber (ID: 22201110)*, Sawda Nawar Pink (ID: 22201668)†, Mohammed Irteza Karim (ID: 20101198)‡
Department of Computer Science and Engineering
BRAC University

***Abstract*—Predictive maintenance aims to detect potential machine failures early to reduce downtime, prevent costly breakdowns, and improve industrial safety. This paper presents an end-to-end machine learning pipeline for failure prediction using industrial sensor and operational data. The workflow includes exploratory data analysis (EDA), missing-value verification and handling strategy, categorical encoding, feature scaling, class imbalance detection and correction using SMOTE, outlier treatment, redundancy removal through correlation-based feature dropping, and feature selection using tree-based feature importance. Seven classification models are evaluated using stratified cross-validation and test-set evaluation with emphasis on ROC–AUC and PR–AUC, which are more informative for imbalanced classification. Results show that ensemble methods dominate classical baselines, and Random Forest achieves the most consistent performance, with near-perfect ROC–AUC and PR–AUC, and the most balanced error profile in tuned evaluation.**

***Index Terms*—Predictive Maintenance, Industrial IoT, Machine Learning, SMOTE, Feature Selection, Random Forest, ROC–AUC, PR–AUC**

## I. Introduction

Industrial environments increasingly rely on continuous, reliable machine operation. Unexpected failures can cause production downtime, reduce throughput, degrade product quality, and in some cases create safety hazards. Predictive maintenance (PdM) addresses these risks by learning patterns from sensor and operational data to predict failures before they occur, enabling planned maintenance and better resource allocation.

In practice, PdM datasets present multiple challenges:

- **Class imbalance:** failures are rare compared to normal operation.
- **Noise and variability:** sensor readings are affected by operational conditions, calibration issues, and environment.
- **Feature redundancy:** many sensor variables can be correlated or partially redundant.
- **Deployment constraints:** model interpretability, computational cost, and reliability matter for industrial adoption.

To address these issues, we implement a structured machine learning workflow that emphasizes reliable preprocessing, imbalance handling, robust evaluation, and model comparison. The final goal is not only high accuracy, but stable generalization across folds and strong performance on minority-class detection.

## II. Related Work (Literature Review)

Predictive maintenance research spans classical machine learning, deep learning, hybrid modeling, and system-level frameworks. Ahmad et al. reviewed application-wise adoption of PdM across manufacturing, energy, transportation, HVAC, and healthcare, noting that imbalance, data scarcity, heterogeneity, and interpretability remain key challenges; they also report strong performance of ensemble and hybrid methods in practice [1].

Deep learning methods have been proposed to capture complex temporal-spatial patterns in sensor streams. Putra et al. used CNN–LSTM architectures on multivariate sensor data in smart manufacturing and reported strong results and practical deployment value, while also implying higher data and computation requirements [2].

Several studies show that robust ensembles can outperform complex models in real industrial settings. Kumar et al. demonstrated Random Forest effectiveness on noisy legacy grinding machine data, showing that robust feature extraction plus ensembles can support cost-effective PdM for aging equipment [3]. Taşcı et al. evaluated RUL prediction in a real factory environment and observed that Random Forest achieved superior performance compared to deeper models under messy, noisy industrial data conditions [5].

More advanced architectures have also been explored. Qi et al. proposed an IIoT framework combining identity resolution and Transformer models, highlighting improved long-term dependency modeling and device tracking for PdM [4]. Security and privacy concerns motivate distributed learning; Hosni proposed Federated Learning for industrial PdM, reducing bandwidth and improving resilience to cyber threats while maintaining predictive quality [6].

Broader surveys reinforce practical challenges and implementation gaps. An NLP-assisted review discussed signal processing, hybrid models, and deployment barriers in industrial PdM [7]. A data-mining survey reviewed methods for industrial asset maintenance and highlighted issues of labeling, variability, and workflow integration [8]. A compressor monitoring study showed that even comparatively simple ML pipelines can deliver real improvements if the end-to-end system is designed carefully [9].

Overall, the literature consistently suggests that (i) data preparation is critical, (ii) imbalance-aware evaluation is nec-

essary, and (iii) ensemble models remain strong baselines in practical industrial datasets. These points directly motivate the workflow and model selection strategy in this project.

## III. Dataset Overview and EDA

The dataset consists of industrial sensor and operational variables such as temperature measurements, rotational speed, torque, tool wear, and a categorical *Type* feature representing machine/operational categories. The target is a binary label indicating failure vs. non-failure.

### A. Target Imbalance in Raw Data

Initial EDA focused on understanding target distribution and whether imbalance handling is required. Figure 1 shows that failures occur far less frequently than normal samples, which can bias models toward predicting the majority class.

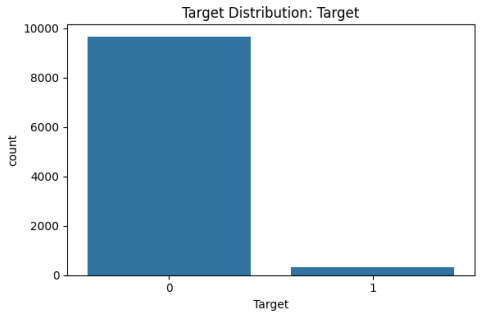



Fig. 1: Target class distribution before applying imbalance handling techniques.

### B. Categorical Feature Distribution

EDA also examined the categorical feature distribution. Figure 2 shows the frequency of each category. This is important because strongly dominant categories can influence learned decision boundaries, and it motivates careful encoding and stratified evaluation.

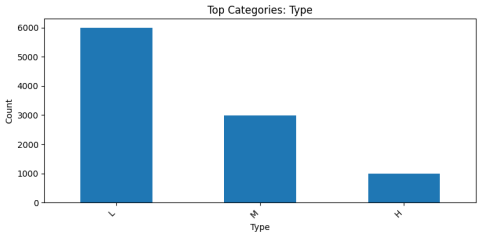



Fig. 2: Distribution of the categorical feature values (Type).

## IV. Methodology

This section describes the complete pipeline implemented in code, including preprocessing, feature selection, model training, and evaluation.

### A. Missing Value Handling

The code first checks missing values column-wise. In our dataset, missing values were not present (or negligible). In general, the pipeline supports standard strategies:

- Mean imputation for approximately symmetric numerical variables,
- Median imputation for skewed numerical variables and robust handling,
- Mode imputation for categorical variables.

Even if missing values are absent, documenting this check is important for reproducibility and ensures the pipeline remains valid when applied to new industrial logs.

### B. Outlier Treatment

Industrial sensor data often contains spikes due to sensor noise or transient events. The pipeline includes outlier treatment before model training. A standard approach is to apply an IQR-based cap (winsorization) or Z-score thresholding for extreme values. Outlier control improves stability, especially for distance-based models such as KNN and margin-based models such as SVM.

### C. Categorical Encoding

Categorical features (e.g., *Type*) cannot be used directly by most ML models. The code converts categorical variables into numerical form using encoding (e.g., one-hot encoding or label encoding, depending on implementation). This step prevents errors such as "could not convert string to float" during training or resampling.

### D. Feature Scaling

Scaling was applied to numerical features using standardization (zero mean, unit variance). This is crucial for models that rely on distances or gradient-based optimization, including SVM and KNN. While tree-based ensembles are less sensitive to scaling, a consistent scaling pipeline allows fair comparisons across models.

### E. Class Imbalance Handling with SMOTE

Because failures are rare, training directly on the raw data can result in high accuracy but poor minority-class detection. The code detects imbalance using the minority-to-majority ratio and applies SMOTE if the ratio is below a threshold.

Figure 3 shows the class distribution before and after SMOTE. After resampling, both classes become balanced, allowing the learning algorithms to observe failure patterns more frequently.

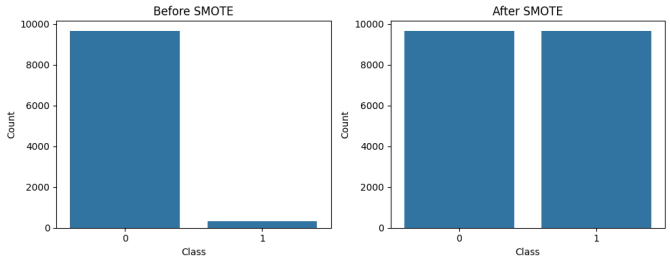



Fig. 3: Class distribution before and after applying SMOTE.

### F. Feature Redundancy Removal

Industrial sensors can be correlated (e.g., temperature-related variables, torque and rotational dynamics). Highly correlated features can cause redundancy and may inflate the importance of certain signals. The code uses an inter-feature correlation threshold to drop redundant features, keeping one representative feature from each highly correlated group. This improves generalization and reduces the risk of overfitting.

### G. Feature Selection via Feature Importance

After removing redundancy, the pipeline fits a Random Forest classifier and uses feature importance to rank the remaining features. The top-$k$ features (e.g., top 7) are selected for final model training. This selection is practical because tree ensembles can capture non-linear interactions and provide a straightforward importance ranking.

## V. Models and Experimental Setup

### A. Models Evaluated

Seven models were trained and compared:

- Logistic Regression (linear baseline),
- SVM with RBF kernel (non-linear margin model),
- K-Nearest Neighbors (distance-based),
- Decision Tree (interpretable baseline tree),
- Naive Bayes (probabilistic baseline),
- Random Forest (bagging ensemble),
- Gradient Boosting (boosting ensemble).

### B. Cross-Validation Strategy

Stratified cross-validation was used to preserve class proportions across folds. This ensures evaluation is not biased by folds that accidentally contain very few failure examples. The code computes mean cross-validation metrics for each model.

### C. Evaluation Metrics

Multiple metrics were tracked:

- Accuracy: overall correctness, can be misleading for imbalance.
- Precision: how many predicted failures are actually failures.
- Recall: how many actual failures are detected (critical in PdM).
- F1-score: balance of precision and recall.
- ROC–AUC: ranking quality across thresholds.
- PR–AUC: more informative than ROC–AUC under imbalance.

Given the PdM context, recall and PR–AUC are especially important because missing failures can be costly.

## VI. Results and Discussion

This section summarizes model behavior using the test confusion-matrix values (reported in generated outputs), cross-validation trends, ROC curves, and tuned model comparison.

### A. Interpreting Confusion-Matrix Behavior (Theory-Based)

Instead of including seven separate confusion-matrix figures, we analyze the misclassification patterns.

**Logistic Regression** produced a large number of false positives and false negatives (e.g., FP = 388 and FN = 344), indicating that a linear boundary is insufficient for complex industrial sensor relationships.

**Naive Bayes** showed very low false positives (FP = 4) but high false negatives (FN = 159). This implies the model is conservative in predicting failures and misses many real failures, consistent with its strong independence assumption that is rarely valid in sensor data.

**Decision Tree** reduced errors compared to linear baselines but still produced moderate false positives (FP = 33) while maintaining low false negatives (FN = 24). Single trees remain sensitive to noise and training variance.

**SVM (RBF)** performed strongly overall but still exhibited more false negatives than the best ensemble (FN = 86). Kernel methods can be sensitive to parameter selection and noisy feature distributions.

**KNN** achieved low false negatives (FN = 31) but had higher false positives (FP = 48), consistent with distance-based confusion in overlapping regions of feature space.

**Gradient Boosting** provided strong performance (FP = 29, FN = 63). Its sequential learning improves prediction but can remain sensitive to residual noise or minor distribution shifts.

**Random Forest** achieved the most balanced behavior among base models (FP = 22, FN = 24). This reflects the advantage of aggregating decorrelated trees, reducing variance and improving robustness.

### B. ROC–AUC Comparison Across Models

Figure 4 compares ROC curves across all models. Random Forest achieved the highest ROC–AUC (approximately 0.999), followed by Gradient Boosting and KNN. Logistic Regression had the lowest curve among compared models, consistent with its weak non-linear capacity.

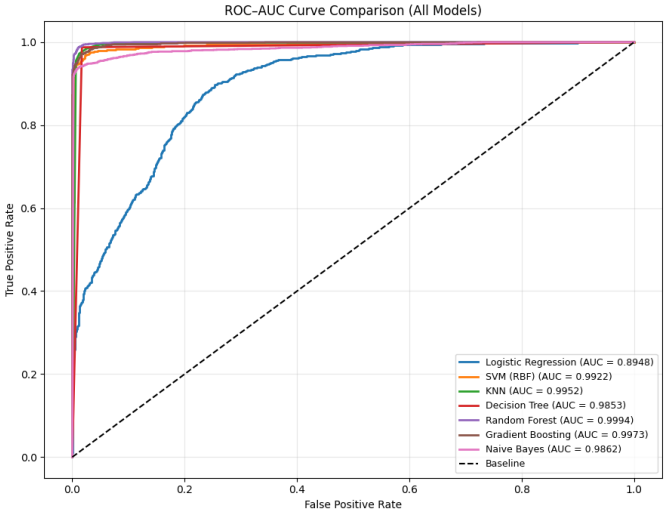



Fig. 4: ROC–AUC curve comparison of all evaluated models.

### C. Cross-Validation and Test Performance Table

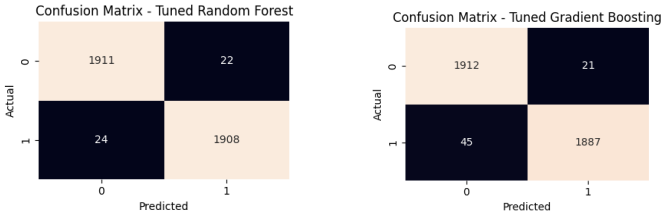
Figure 5 summarizes model comparison using cross-validation accuracy/precision/recall/F1 and ROC–AUC/PR–AUC, alongside test ROC–AUC and test PR–AUC. The ranking is primarily based on PR–AUC because it better captures minority-class performance in imbalanced settings.

| | Model | CV Accuracy | CV Precision | CV Recall | CV F1 | CV ROC-AUC | CV PR-AUC | Test ROC-AUC | Test PR-AUC |
|---|---|---|---|---|---|---|---|---|---|
| 4 | Random Forest | 0.986479 | 0.989335 | 0.983568 | 0.986436 | 0.999339 | 0.999357 | 0.999388 | 0.999395 |
| 5 | Gradient Boosting | 0.975415 | 0.985089 | 0.965454 | 0.975165 | 0.997520 | 0.997695 | 0.997336 | 0.997678 |
| 1 | SVM (RBF) | 0.970175 | 0.986897 | 0.953033 | 0.969654 | 0.992025 | 0.992416 | 0.992214 | 0.991604 |
| 2 | KNN | 0.976774 | 0.972805 | 0.980981 | 0.976872 | 0.994922 | 0.992027 | 0.995195 | 0.991792 |
| 6 | Naive Bayes | 0.955295 | 0.998010 | 0.912407 | 0.953282 | 0.986374 | 0.988494 | 0.986158 | 0.988467 |
| 3 | Decision Tree | 0.981173 | 0.980257 | 0.982145 | 0.981190 | 0.981173 | 0.971680 | 0.985253 | 0.976997 |
| 0 | Logistic Regression | 0.811024 | 0.797953 | 0.833223 | 0.815115 | 0.897573 | 0.893443 | 0.894752 | 0.888052 |

Fig. 5: Cross-validation and test performance comparison of evaluated models.

### D. Hyperparameter Tuning

Hyperparameter tuning was applied to the two strongest candidates: Random Forest and Gradient Boosting. Figure 6 shows the tuned confusion matrices. Tuned Random Forest remained highly stable and balanced (very low FP and FN), confirming it as the final selected model.




(a) Tuned Random Forest (b) Tuned Gradient Boosting

Fig. 6: Confusion matrices of the two best tuned models.

### E. Why Random Forest Was Selected

Random Forest was selected because it:

- Achieved the highest ROC–AUC and near-perfect PR–AUC,
- Produced one of the most balanced error profiles (low FP and low FN),
- Was stable across cross-validation folds and test evaluation,
- Is robust to noise and partially redundant industrial sensor variables.

These findings match multiple industrial PdM studies that report strong ensemble robustness under real operational noise [1], [3], [5].

## VII. Conclusion and Future Work

This project implemented a full predictive maintenance pipeline on industrial and machine sensor data. The workflow included systematic EDA, preprocessing, SMOTE-based imbalance correction, feature redundancy removal, feature selection, and evaluation of seven models with stratified cross-validation. Ensemble methods clearly outperformed simpler baselines, and Random Forest emerged as the best overall classifier based on ROC–AUC, PR–AUC, and balanced error behavior.

Future work can extend this project by:

- Exploring explainable AI techniques to improve operator trust (e.g., SHAP/LIME),
- Testing time-aware models when temporal order is available (e.g., LSTM/Transformers),
- Evaluating deployment strategies such as edge inference for low-latency PdM,
- Investigating privacy-preserving training via Federated Learning [6].

## References

[1] M. Ahmad, A. Rahman, and S. Islam, "Application-wise review of machine learning-based predictive maintenance: Trends, challenges, and future directions," *Applied Sciences*, vol. 15, no. 9, p. 4898, 2024.

[2] A. R. Putra, H. Wijaya, and B. Santoso, "AI-driven predictive maintenance for smart manufacturing systems: A case study using deep learning on sensor data," *Technik Journal*, 2023.

[3] R. Kumar, P. Singh, and S. Verma, "Predictive maintenance of old grinding machines using machine learning techniques," *Journal of Applied Engineering and Technological Science*, 2023.

[4] Y. Qi, L. Zhang, and H. Wang, "Predictive maintenance based on identity resolution and transformers in IIoT," *Future Internet*, vol. 16, no. 9, p. 310, 2024.

[5] E. Taşcı, B. Demir, and A. Yıldız, "Remaining useful lifetime prediction for predictive maintenance in manufacturing," *Computers in Industry*, vol. 149, p. 103951, 2023.

[6] M. Hosni, "Secure predictive maintenance for industrial systems using federated learning," 2026.

[7] T. Zhang, Y. Liu, and X. Chen, "Condition monitoring and predictive maintenance in industrial equipment: An NLP-assisted review," *Applied Sciences*, vol. 15, no. 10, p. 5465, 2024.

[8] F. García, J. López, and A. Ruiz, "A survey on data mining for data-driven industrial assets maintenance," *Risks*, vol. 13, no. 2, p. 67, 2024.

[9] D. Martinez, R. Silva, and P. Costa, "Machine learning implementation to predictive maintenance and monitoring of industrial compressors," *Sensors*, vol. 25, no. 4, p. 1006, 2024.