

Machine Learning-Based Predictive Maintenance Classification for Industrial Equipment Failure Detection

Aravindh Balan

Freelance Postdoctoral Scholar

Project Manager

Inline Hydraulics GmbH, Germany

draravindhbalan@gmail.com

Abstract—Industrial equipment optimization is crucial in modern industries for boosting operating efficiency, decreasing downtime, and minimizing maintenance costs. Predictive maintenance helps achieve these goals by predicting equipment failures and scheduling repairs using machine learning. For the purpose of detecting machine failures, this study employs machine learning-based predictive maintenance using operational data from a subset of industrial equipment included in the AI4I 2020 Predictive Maintenance dataset. Data is cleaned, outliers are detected and removed, One-Hot Encoding is applied, and class balancing is achieved utilizing SMOTE techniques to improve data quality, generate a more balanced class distribution, and improve the performance of the predictive model. Classification of industrial equipment failures was investigated using a variety of models, including Decision Tree (DT), Logistic Regression (LR), CNN, K-Nearest Neighbors (KNN), Random Forest and XGBoost. The testing findings showed that out of all the models tested, XGBoost achieved 98.4% accuracy and Random Forest 98.3%. Overall, the results showed the high accuracy of predictive maintenance and detection of industrial equipment failures that can be achieved using advanced ensemble and deep learning methods.

Keywords—Machine Learning, Industrial Machinery, Predictive Maintenance, Condition Monitoring, Failure Prediction.

I. INTRODUCTION

In manufacturing, power generation and transportation systems, where continuity and reliability of operations are critical for productivity and cost efficiency, information and communication technology (ICT) plays an important role [1]. Any unexpected machinery failures can have severe implications such as production loss, high maintenance expenses, safety issues, and supply chain disruption [2]. Reactive maintenance (RM) is a strategy that is implemented in response to a breakdown, while preventive maintenance (PM) is maintenance that is conducted at regular intervals or times of usage [3]. However, traditional methods are often not optimal for the use of resources and maintenance activities, which is not desirable for the modern data-driven industrial world [4].

To address these issues, industries are increasingly moving towards more intelligent and proactive maintenance strategies [5]. The reliability of equipment will contribute to the quality of production, safety of operations and system performance [6]. Unexpected failures lead to higher cost of operation, lower productivity, and could be dangerous to human operators [7][8]. In most of the traditional systems, the maintenance work is still being done with the traditional methods which lead to longer and more frequent downtime periods and inefficient decision making [9].

Predictive maintenance (PdM) is a contemporary method that uses historical data analysis and real-time equipment monitoring to predict failure. It has emerged as a new paradigm [10]. It is designed to combine IIoT, sensor technologies and data-driven analysis for continuous monitoring of machine conditions such as vibration, torque,

temperature, tool wear and rotational speed [11]. Product lifecycle management (PdM) aids businesses in reducing unscheduled downtime and improving maintenance planning by assisting with remaining usable life (RUL) prediction and proactive failure symptom detection [12]. Deep Learning (DL) and ML are the backbone of modern Predictive Maintenance systems, allowing for intelligent analysis of large datasets generated in industrial applications in order to detect and classify failures [13]. The ML-based PdM adopts supervised and unsupervised learning techniques, with supervised learning algorithms making use of historical data with labels for classification or regression problems in fault prediction [8]. Using these models on operational data, it is possible to optimize the maintenance schedule, avoid redundant inspections, prolong the life of machines and maximize return on investment [14].

A. Motivation and Contribution

This study is driven by the greater need to prevent any unexpected failure of industrial equipment and to eliminate unnecessary and expensive downtime in contemporary industries. Traditional maintenance approaches are often inefficient and fail to provide early failure prediction in dynamic industrial environments. The availability of data collected from industrial sensors presents a great chance to improve decision-making via the use of smart, data-driven methods. Therefore, this research primarily aims to create a reliable predictive maintenance system that can accurately detect faults through the use of ML models. This research offers several key contributions as listed below:

- Utilized the AI4I 2020 predictive maintenance data for industrial equipment failure detection.

- Applied pre-processing techniques including outlier removal, One-Hot Encoding, Z-score normalization, and SMOTE-based data balancing.
- Implemented and evaluated advanced predictive maintenance models, including XGBoost and Random Forest, for industrial equipment failure detection and performance analysis.
- Evaluated model performance using accuracy, recall, precision, confusion matrix, F1-score, and ROC curve analysis.
- Developed an optimized predictive maintenance framework for accurate industrial failure classification.

B. Justification and Novelty

This investigation is justified by the increasing significance of dependable predictive maintenance systems in reducing industrial equipment outages and improving operational efficiency. The novelty of this study lies in the implementation and comparison of advanced ensemble learning models, namely XGBoost and Random Forest, for industrial equipment failure detection using the AI4I 2020 predictive maintenance dataset. The study incorporates pre-processing techniques, including outlier removal, One-Hot Encoding, Z-score normalization, and SMOTE-based data balancing to enhance prediction performance. In addition, including rotational speed, torque, mechanical power, and tool wear, for accurate predictive maintenance classification.

C. Organization of the Paper

The following is the paper's structure: A literature overview on industrial equipment failure detection is provided in Section II, followed by an explanation of the dataset, preprocessing, and model implementation in Section III. Results and comparative analysis are presented in Section IV, and the study is concluded with important conclusions and recommendations for future research in Section V.

II. LITERATURE REVIEW

A comprehensive review of key studies on industrial equipment failure detection is conducted to support this work. A summary of the recent research in this area is provided in Table I, including the proposed models, the datasets employed, the major results, and the limitations.

Kadirvel et al. (2026) used both hybrid deep learning models and simplified communication protocols to make things more efficient, reliable, and able to forecast when maintenance is needed. The suggested system uses a multi-

layer Digital Twin to keep physical industrial assets in sync all the time. It has a real-time health score of 92 and a minimal synchronization delay of 12ms. can find flaws with 96% accuracy and anticipate failures up to 18 hours in advance using hybrid CNN-LSTM and anomaly detection models [15].

Pati et al. (2025) trained model on a publicly available dataset from Kaggle that simulates industrial motor conditions. The model effectively predicts motor health based on parameters including temperature, vibration, and load. They also developed a monitoring dashboard to visualize predictions and system health. The system achieves over 96% accuracy and is ready for deployment in real-time industrial applications [16].

Mishra et al. (2025) implemented on packaging film slitting machines with tension, pressure, width, and diameter sensors, it achieved 89% accuracy in predicting failures up to 45 minutes in advance. In industrial settings, this implementation greatly decreases unscheduled downtime and maintenance expenses while allowing enough time for remedial measures [17].

Wadibhasme et al. (2024) aim to highlight the ongoing imperative for innovative and adaptable practices that strive for operational excellence. In performance comparisons for predictive maintenance, Neural Networks achieved the highest metrics: 96.3% accuracy, 97.2% recall, 94.8% precision and Random Forests followed with 94.5% accuracy, 92.7% precision, 95.8% recall, and a 94.2% F1 score. Decision Trees achieved 92.1% accuracy, 89.3% precision, 94.2% recall, and 91.7% F1 score, while Logistic Regression had the lowest performance with 85.2% accuracy, 82.6% precision, 87.5% recall, and an 84.8% F1 score [18]. Choi and Im (2024) assessed using metrics such as F1 score, AUC, and accuracy. To get the best threshold value, the evaluation method uses the Youden index. In order to find the bandwidth of the Gaussian kernel used in MKDE, various existing approaches are applied to evaluate its performance. The MKDE model developed through the experiment shows an AUC of 0.9663, an accuracy of 0.8971, and an F1score of 0.9041 [19].

Praveena et al. (2024) suggested framework is a major step forward in industrial machinery predictive maintenance, providing a scalable, accurate, and privacy-preserving approach for proactive failure identification and prediction. Its use of hybrid MLP-GRU model architecture and federated learning approaches shows promising outcomes of 0.94% accuracy in practical industrial applications [20]

TABLE I. COMPARATIVE ANALYSIS OF RECENT MACHINE LEARNING-BASED INDUSTRIAL EQUIPMENT FAILURE DETECTION

Author	Proposed Work	Dataset		Limitations & Future Work
Kadirvel et al. (2026)	Hybrid Digital Twin framework using CNN-LSTM and anomaly detection for predictive maintenance	Industrial Digital Twin system data	96% accuracy, 92 health score, 18-hour failure prediction, 12 ms sync delay	High computational cost; future work on lightweight real-time deployment
Pati, Zope and Bhole (2025)	Machine learning-based motor health monitoring system with dashboard integration	Kaggle industrial motor dataset	>96% accuracy with real-time monitoring dashboard	Limited real-world validation; future work on scalability and sensor fusion
Mishra et al. (2025)	Predictive maintenance system for packaging film slitting machines using sensor data	Industrial machine sensor dataset	89% accuracy, failure prediction up to 45 minutes	Limited accuracy; future improvement in early fault detection
Wadibhasme et al. (2024)	Comparative study of ML models for predictive maintenance	Industrial maintenance datasets	Neural Networks: 96.3% accuracy, RF: 94.5%, DT: 92.1%	Focus only on comparison; lacks real-time deployment analysis
Choi and Im (2024)	MKDE-based fault detection model with ROC and AUC evaluation	Industrial process dataset	0.9663 AUC, 89.71% accuracy, 0.9041 F1-score	Lower performance vs deep learning; needs kernel optimization

Praveena et al. (2024)	Federated learning with hybrid MLP-GRU for predictive maintenance	Industrial dataset	IoT	~94% accuracy with privacy-preserving learning	Communication overhead in federated learning
------------------------	---	--------------------	-----	--	--

Research gaps: The existing literature on industrial equipment failure detection has made strong progress in applying ML and DL techniques; however, several gaps remain. Various research has only focused on single models or just a few models for comparison, and have not been conducted to determine how effective models are in different scenarios. Furthermore, many of the methods have been evaluated on small or specific data sets, making it difficult to

see how well they generalize to real-world industrial settings. There are some works that do not cover the problem, such as real-time deployment constraints, class imbalance and computational efficiency. Therefore, there was a need for more powerful, scalable, and balanced models to improve the accuracy, generalization and use in the industrial predictive maintenance systems.

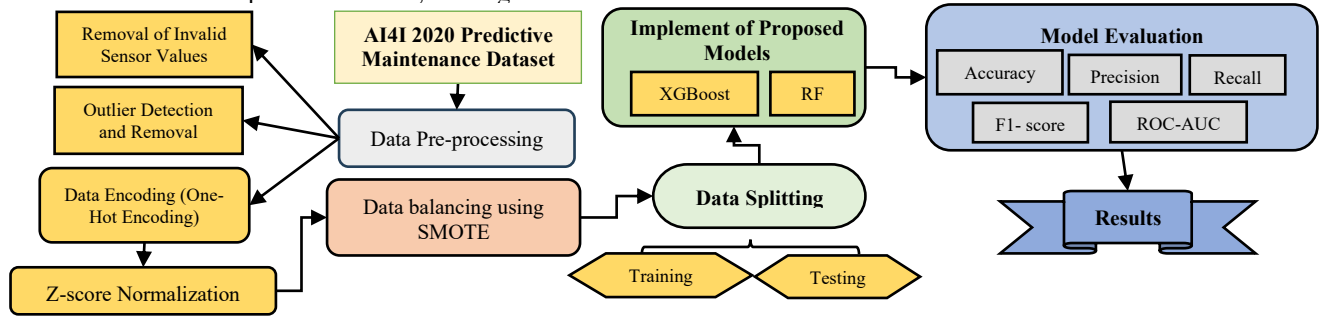


Fig. 1. Proposed flowchart for Industrial Equipment Failure Detection

III. RESEARCH METHODOLOGY

In this research, approach to failure detection of industrial equipment based on the AI4I 2020 data is implemented, including data collection, data preprocessing and data balancing by SMOTE. The most important operational features of the equipment are fed to ML models, including RF and XGBoost, for the classification of equipment failure patterns. Recall, accuracy, precision, F1score, and ROC curve analysis are some of the metrics utilized to assess model performance. The proposed flowchart for the detection of failure of industrial equipment is presented in Fig. 1.

The next part provides a full description of each phase in the proposed methodology:

A. Data Gathering and Analysis

The AI4I 2020 Predictive Maintenance dataset, which is available on Kaggle [21] This is the basis for this study. It is a synthetic industrial dataset with 10,000 records and 14 features that represent the operating conditions of a milling machine. Factors such as process temperature, air temperature, tool wear, torque, and machine failure are among the data points included in the gathered information. This makes it a good fit for industrial machinery failure detection and predictive maintenance.

Data visualization methods are used to analyze class distributions, feature relationships and feature correlations between variables as shown below:

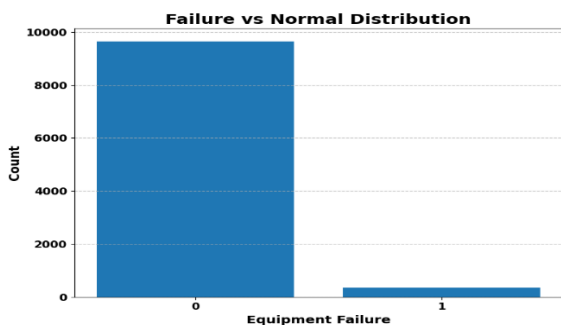


Fig. 2. Bar Graph of Class Distribution

Fig. 2 illustrates the class distribution of normal and failure in the data set. The normal class is seen to have many more samples than the failure class, which shows an imbalance in the distribution of samples in the dataset.

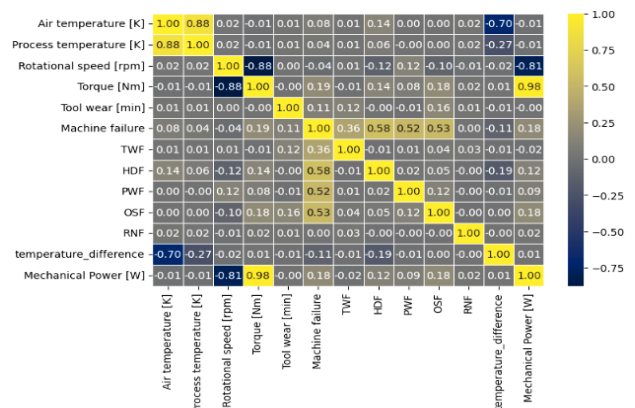


Fig. 3. Correlation Matrix Heatmap

Fig. 3 shows the correlations among features in the data set using correlation coefficients. There are a number of variables that are strongly correlated, which helps to explain feature dependencies and what effect these features have on machine failure prediction.

B. Data Pre-processing

Data preparation includes data integration, cleaning and feature-importance analysis using the “AI4I 2020 Predictive Maintenance Dataset”. In the preprocessing phase, the data quality is enhanced by identifying and discarding outliers and data labeling and normalization approaches were utilized to advance the data quality and performance of the model. The main processing operations are briefly described below.

- **Removal of Invalid Sensor Values:** Invalid machine readings like negative rotational speed or torque values, are detected as they do not meet the physical operating limits of industrial machines.
- **Outlier Detection and Removal:** Any outliers are detect utilized statistic approaches such as Interquartile

Range (IQR) analysis. Extreme values are removed to minimize noise and increase the reliability of the model.

C. Data Encoding Using One-Hot Encoding

The AI4I 2020 Predictive Maintenance Dataset heavily employs One-Hot Encoding for the purpose of numerically representing category data. The data is composed of various categorical features, including equipment types and failure modes of the machines. For machine learning models to learn and make predictions from these unique attributes, each category is transformed into a binary column using one-hot encoding.

D. Data Normalization using Z-score Normalization

As a preprocessing step, data normalization ensures that all of the data points are uniform in size and distribution. In this study, the data is altered to have a mean of 0 and a Standard Deviation of 1 using z-score normalization. This technique makes the feature values uniformly distributed around the mean. As seen in Equation (1), z-score normalization is mathematically represented.

$$E' = \frac{E - \bar{M}}{\sigma_M} \quad (1)$$

where

σ_M is the standard deviation, \bar{M} as the mean, E' and E are new and old for each data entry, and

E. Data Balancing using SMOTE

Data balancing is the process of addressing an imbalanced dataset, where one class has significantly fewer samples than the others, by adjusting the distribution of classes to advance the ML model's performance. The data shows that there are too many classes because machines don't break down very often. This could cause models to make overly optimistic predictions. To address this problem, the training set is augmented using the Synthetic Minority Oversampling Technique (SMOTE). SMOTE generates artificial specimens of the minority failed class by interpolating among the current examples, thus balancing class distributions. This ensures fair learning and advances the accuracy of models in predicting rare failures.

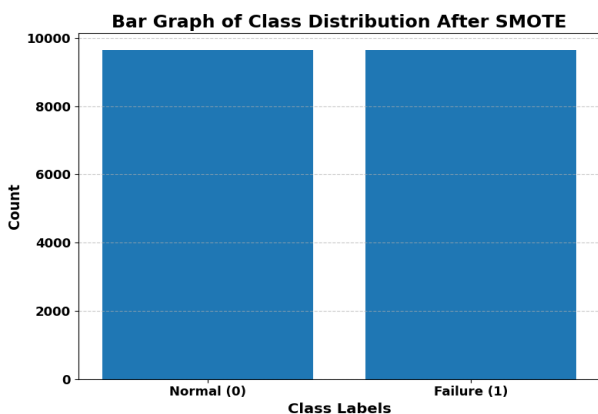


Fig. 4. Bar Graph of Class Distribution After SMOTE

Fig. 4 shows the balanced distribution of the normal and failure classes after performing SMOTE technique. The synthetic oversampling process is used to enhance the minority class, creating an equal-size class for the minority class.

F. Data Splitting

The train-test split is performed in a stratified manner with an 80:20 ratio. This stratified split is used to preserve the training and testing sets' proportionate class distributions, which mirror those of the original dataset.

G. Implement of Proposed Models

In this study, the proposed models, including RF and XGBoost, are implemented for industrial equipment failure detection.

1) Proposed Extreme Gradient Boosting (XGBoost) Model

An ensemble learning technique that uses decision trees is XGBoost. Regression problems may be solved by minimizing a loss function that measures the margin of error between the target and predicted values. The XGBoost regression mathematical model is Equation (2):

$$y = f(x) \quad (2)$$

In this context, x represents the input feature vector (such as number of bedrooms or square footage), $f(x)$ is the XGBoost model that uses x to predict y , and y is the expected property price. To determine $f(x)$, XGBoost compiles a set of decision trees that have been trained to minimize the loss function known as MSE. In order to make a final forecast, the model takes the output of numerous decision trees and merges them. As a whole, the XGBoost regression model looks like Equation (3):

$$y = \sum_{k=1}^K f_k(x) \quad (3)$$

where K is the sum of all the ensemble's decision trees and $f_k(x)$ is their output for the k -th decision tree. The training process teaches each tree its leaf values, and the prediction is just a weighted average of those values. After adding up the forecasts of each decision tree in the ensemble, we get the XGBoost model's prediction for a certain input x .

Here are the hyperparameters for the XGBoost model: `min_child_weight = 1`, `n_estimators = 200`, `max_depth = 6`, `learning_rate = 0.1`, `subsample = 0.8`, and `colsample_bytree = 0.8`, `gamma = 0`. These parameters are selected to avoid overfitting and optimize the model's complexity for use in industrial equipment failure detection prediction.

2) Random Forest (RF) Model

Random Forest improves regression or classification performance by building and merging several decision trees. It is an ensemble machine learning approach. A method to enhance diversity among the trees is to train them on randomly selected portions of the training data (bagging) and to employ randomly selected features at each split. Classification uses majority voting, and regression uses average across all trees, to arrive at the final forecast. Sections are created for the data in a recursive fashion. Splitting occurs at each node when an attribute is queried. The splitting criterion is chosen according to certain impurity metrics, such as the Shannon entropy or Gini impurity. Using the Gini impurity function, we can measure the split quality of every node. For node N , the Gini impurity can be calculated by applying Equation (4).

$$g(N) = \sum_{i \neq j} P(w_i)P(w_j) \quad (4)$$

where $P(w_i)$ is the percentage of people who have class label i . An additional measure that may be used to assess the split's quality is the Shannon entropy. It gives a numerical value to the data's disorder. For decision trees, the Shannon entropy indicates how unpredictable the information stored

inside a node is due to the population mixing at that node. In Equation (5), the entropy in node N can be determined in the following way:

$$H(N) = \sum_{i=1}^d P(w_i) \log_2(Pw_i) \quad (5)$$

where $P(w_i)$ is the percentage of the population identified as i and d is the total number of categories analyzed. Entropy is highest when every class is dispersed equally throughout the node. It is the lowest when a node contains only one class (pure). The most basic criterion for choosing the optimal node splitting is to reduce impurity. You may say that the optimal split is defined by the greatest improvement in data or the minimum amount of contaminants. Equation (6) allows us to estimate the information benefit from a split to be:

$$\Delta I(N) = I(N) - P_L * I(N_L) - P_R * I(N_R) \quad (6)$$

where P_R is the percentage of the population in node N that goes to the right child following the split, P_L is the percentage of the population in node N that goes to the left child of N following the split, and $I(N)$ is the impurity measure (Gini or Shannon Entropy) of node N. N_L and N_R are the left and right children of N, respectively. By default, the Random Forest model's hyperparameters are set to 42 for random state, 15 for max depth, 1 for n_min_samples_leaf, 2 for min_samples_split, True for bootstrap, and sqrt for max features. With these parameters, we can construct more robust decision trees, which in turn improve our classification performance and generalizability.

H. Evaluation Metrics

The suggested layout is assessed using a number of performance indicators. To start, the results of the categorization, broken down by class and including the number of right and wrong guesses, are shown in a confusion matrix. Derived from this matrix are True Negatives (TN), False Negatives (FN), True Positives (TP) and False Positives (FP). The following are the main evaluation metrics that are computed using these values: precision, recall, accuracy, and F1 score.

Accuracy: It is the proportion of instances correctly forecasted by the trained model to the number of occurrences in the data set. It is expressed as Equation (7)-

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

Precision: It is the proportion of correct positive forecasting to the total number of positive forecasting produced by the model. It is the rate of correct classification of the positive classes and is expressed as Equation (8)-

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Recall: The ratio of correctly identified positive cases to all positive cases is raised to a certain threshold. The accuracy with which the model can detect positive instances is measured by this mathematical performance parameter. And it is mathematically expressed as Equation (9)-

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

F1 score: It is a hybrid of recall and precision that allows for a more precise evaluation. Improved model performance is indicated by higher numbers, which might range from 0 to 1. The mathematical expression is Equation (10)-

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

Receiver Operating Characteristic Curve (ROC): It is the plot of the FPR versus TPR with respect to various thresholds. TPR is also referred to as sensitivity or recall, while FPR is 1-specificity.

IV. RESULTS AND DISCUSSION

The experimental setup and procedure to implement the proposed model for the early prediction of fault is described in this section. It also reports on empirical experiments conducted and an extended discussion on the performance and effectiveness of the model.

A. Experimental Setup

For model development and data processing, the software requirements are Jupyter Notebook, NumPy, Python 3, Seaborn, Matplotlib, Pandas, and Scikit-learn. The hardware specifications include a system with sufficient memory (16 GB or more) and the capability to effectively analyze massive datasets and intricate machine learning algorithms [22].

B. Performance Evaluation Results

As shown in Table II, the software requirements for model development and data processing, the software requirements are Jupyter Notebook, Python 3, Pandas, estimators = 300, NumPy, Scikit-learn, Matplotlib and Seaborn. The required hardware consists of a regular desktop or laptop computer with enough processing speed and memory (16 GB or more) to properly manage big data sets and complicated ML algorithms. With an accuracy of 98.4%, the XGBoost model outperformed the Random Forest model, which had an accuracy of 98.3%. It also achieved a better recall of 58.5% and an F1 of 71.3%. The results have shown that the XGBoost model is more balanced and accurate at detecting failures in industrial equipment.

TABLE II. EXPERIMENTAL RESULTS OF THE PROPOSED MODEL FOR INDUSTRIAL EQUIPMENT FAILURE DETECTION

Matrix	XGBoost	Random Forest
Accuracy	98.40	98.35
Precision	79.69	91.11
Recall	72.86	58.57
F1-score	76.11	71.30

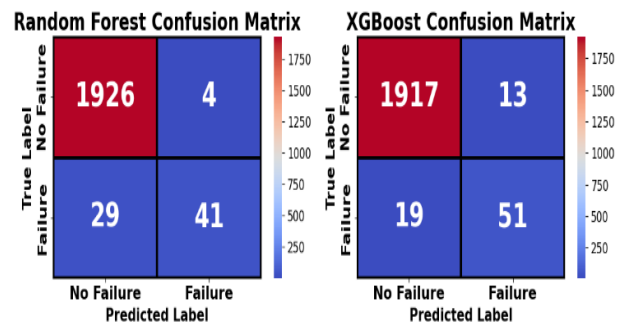


Fig. 5. Confusion Matrix of the Proposed Random Forest and XGBoost Models

Fig. 5 displays the XGBoost and RF models' confusion matrices. The RF model correctly classified 1926 "No Failure" samples and 41 "Failure" samples, with 4 and 29 misclassifications, respectively. Similarly, the XGBoost model correctly identified 1917 "No Failure" samples and 51 "Failure" samples, while 13 normal and 19 failure samples were incorrectly classified.

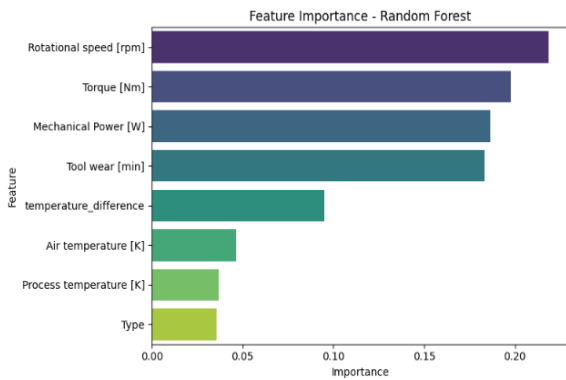


Fig. 6. Feature Importance Analysis of the Proposed Random Forest Model

Fig. 6 illustrates the Feature Importance results of the proposed RF model for Predictive Maintenance classification. Rotational speed [rpm] showed the highest importance, followed by torque [Nm], mechanical power [W], and tool wear [min]. Predictions were less affected by air temperature, process temperature, and machine type.

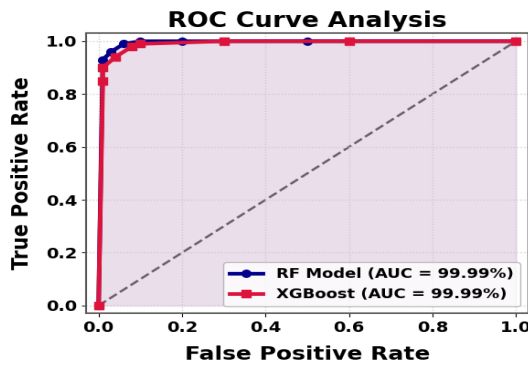


Fig. 7. ROC Curve for proposed Models

Fig. 7 displays the ROC curves for the RF and XGBoost models, which predict machine failures. Both the models obtained a high AUC value of 99.99% for the model Random Forest and 99.99% for the model XGBoost, which showed a good predictive ability and discrimination power between failure and non-failure classes.

C. Comparative Analysis

Table III demonstrates how the proposed models fare by comparing the various classifiers' results with those of preexisting ML models. The tables display the classifiers' accuracy, precision, recall, and F1-scores. Overall, XGBoost outperformed all other models evaluated, achieving an accuracy of 98.4%, while the Random Forest model came in at 98.3%. The CNN and KNN models also performed well and balanced on all the evaluation metrics. The highest precision is achieved by Random Forest (91.1%), but with relatively low recall. However, when compared to the other models, such as the DT and LR models, the advanced models produced superior overall performance.

TABLE III. COMPARISON OF DIFFERENT ML AND DL MODELS FOR INDUSTRIAL EQUIPMENT FAILURE DETECTION

Model	Accuracy	Precision	Recall	F1-score
DT[23]	72.3	79.8	80.1	80.03
LR[24]	91	72	60	65
KNN[25]	97.3	96.2	97.3	96.5
CNN[26]	97.5	96.4	96.8	97.2
Proposed XGBoost	98.40	79.69	72.86	76.11
Proposed RF	98.35	91.11	58.57	71.30

V. CONCLUSION AND FUTURE STUDY

Nowadays, digitalization and automation in industrial processes have become standard across many companies, where maintenance plays a crucial role in minimizing equipment downtime. This study concentrates on Predictive Maintenance as one of the various maintenance strategies. The proposed XGBoost (98.4%) and Random Forest (98.3%) models have the highest accuracy, while traditional models, including Decision Tree (72.3%) and Logistic Regression (91%) show lower accuracy, and the models K-NN (97.3%) and CNN (97.5%) show higher accuracy. This proves the effectiveness and robustness of ensemble-based methods in the prediction of the failure of industrial equipment. This study only focused on a single dataset, which could limit the extent to which conclusions can be drawn for use in other industrial settings. It also emphasizes supervised machine learning models and pays little attention to more challenging real-time or streaming data scenarios. Future research can involve testing the proposed models with real-world industrial data and integrating progressive models such as LSTM, CNN, and Hybrid CNN-LSTM for further enhancing the predictive maintenance accuracy. Ensemble learning, Explain ability (SHAP/LIME), and IoT & Edge computing are additional techniques to advance the accuracy of the failure finding of industrial equipment.

REFERENCES

- [1] A. Abdel-Monem and M. Abouhawwash, "A Machine Learning Solution for Securing the Internet of Things Infrastructures," *Sustain. Mach. Intell. J.*, vol. 1, Oct. 2022, doi: 10.61185/SMIJ.HPA09103.
- [2] S. Das, V. Challagulla, P. Balaramesh, and A. Mohan, "Deep Learning for Corrosion Monitoring Virtual Sensor and Predictive Modelling Approaches in Industrial Water Pipeline," *The Bioscan*, vol. 21, no. 1, pp. 35-45, Jan. 2026, doi: 10.63001/tbs.2026.v21.i01.pp35-45.
- [3] A. Bousdekis, B. Magoutas, D. Apostolou, and G. Mentzas, "Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance," *J. Intell. Manuf.*, vol. 29, no. 6, pp. 1303-1316, Aug. 2018, doi: 10.1007/s10845-015-1179-5.
- [4] S. Doda, N. Kamuni, P. Notalapati, and J. R. Vummadi, "Intelligent Data Processing for IoT Real-Time Analytics and Predictive Modeling," in *2025 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Jul. 2025, pp. 649-654. doi: 10.1109/ICoDSA67155.2025.11157424.
- [5] A. M. Ali and A. Abdelhafeez, "DeepHAR-Net: A Novel Machine Intelligence Approach for Human Activity Recognition from Inertial Sensors," *Sustain. Mach. Intell. J.*, vol. 1, Nov. 2022, doi: 10.61185/SMIJ.2022.8463.
- [6] A. Hatip, K. Zayood, and R. Sharif, "Innovations at the Nexus of Sustainability and Industry 4.0: Data-Driven Approach for Preemptive Equipment Management in Smart Factories," *Int. J. Wirel. Ad Hoc Commun.*, vol. 7, no. 1, pp. 40-49, 2023, doi: 10.54216/IJWAC.070104.
- [7] N. R. Barot, "Transparency-Driven Operational Intelligence: A New Data Governance Model for High-Risk Industrial Automation," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 63s, pp. 1019-1028, Dec. 2025, doi: 10.52783/jisem.v10i63s.13975.
- [8] T. Akyaz and D. Engin, "Machine Learning-Based Predictive Maintenance System for Artificial Yarn Machines," *IEEE Access*, vol. 12, no. September, pp. 125446-125461, 2024, doi: 10.1109/ACCESS.2024.3454548.
- [9] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0," *Sustainability*, vol. 12, no. 19, Oct. 2020, doi: 10.3390/su12198211.
- [10] R. Atassi and F. Alhosban, "Predictive Maintenance in IoT: Early Fault Detection and Failure Prediction in Industrial Equipment," *J.*

- Intell. Syst. Internet Things*, vol. 9, no. 2, pp. 231–238, 2023, doi: 10.54216/JISIoT.090217.
- [11] T. M. Le, H. M. Tran, K. Wang, H. V. Pham, and S. V. T. Dao, “An Internet-of-Things-Integrated Deep Learning Model for Fault Diagnosis in Industrial Rotating Machines,” *IEEE Access*, vol. 13, no. March, pp. 57266–57286, 2025, doi: 10.1109/ACCESS.2025.3553155.
- [12] S. Butte, A. R. Prashanth, and S. Patil, “Machine Learning Based Predictive Maintenance Strategy: A Super Learning Approach with Deep Neural Networks,” in *2018 IEEE Workshop on Microelectronics and Electron Devices (WMED)*, IEEE, Apr. 2018, pp. 1–5. doi: 10.1109/WMED.2018.8360836.
- [13] N. Sahasrabudhe, R. Asegaonkar, S. Deo, S. Umredkar, and K. Mundada, “Experimental Analysis of Machine Learning Algorithms used in Predictive Maintenance,” in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, Aug. 2020, pp. 1302–1308. doi: 10.1109/ICSSIT48917.2020.9214181.
- [14] A. Jain, “AI-Powered Predictive Maintenance Using Deep Learning for Industrial IoT Environments,” vol. 4, no. 6, pp. 5824–5837, 2021, doi: 10.15662/IJARCST.2021.0406009.
- [15] M. Kadirvel, A. W. A. Lenin, J. R. Pagunuran, M. R. Katta, J. E. Krishnankutty, and R. Arumuganainar, “A Multi-Protocol Digital Twin Framework for Intelligent Fault Diagnosis in Industrial IoT Using Hybrid Deep Learning and Low-Power Routing Techniques,” in *2026 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*, IEEE, Mar. 2026, pp. 1747–1753. doi: 10.1109/ICMLAS67792.2026.11483683.
- [16] P. J. Pati, P. H. Zope, and M. K. Bhole, “Predictive Maintenance of Industrial Equipment Failure Using Random Forest Algorithm,” in *2025 2nd International Conference on Integration of Computational Intelligent System (ICICIS)*, IEEE, Sep. 2025, pp. 1–6. doi: 10.1109/ICICIS65613.2025.11371090.
- [17] A. Mishra, A. Tewari, A. Anand, A. Garg, and N. Jain, “Machine Learning for Predictive Maintenance of Industrial Machines using IoT Sensor Data: A Case Study on Slitting Machines,” in *2025 International Conference on Sensors and Related Networks (SENNET) Special Focus on Digital Healthcare(64220)*, IEEE, Jul. 2025, pp. 1–5. doi: 10.1109/SENNET64220.2025.11135975.
- [18] R. N. Wadibhasme, M. Naresh, G. Vikram, A. S. V. V., S. P., and F. Jermina, “Utilizing Machine Learning Techniques for Enhanced Predictive Maintenance in the Manufacturing Sector,” in *2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ICNEWS60873.2024.10730983.
- [19] J. Choi and S. Im, “Anomaly Detection of Three-Phase Induction Motor Based on Unsupervised Learning Using Frequency Distortion Characteristics,” in *2024 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, Jan. 2024, pp. 1–5. doi: 10.1109/ICCE59016.2024.10444181.
- [20] K. Praveena, M. Misba, C. Kaur, M. S. Al Ansari, V. A. Vuyyuru, and S. Muthuperumal, “Hybrid MLP-GRU Federated Learning Framework for Industrial Predictive Maintenance,” in *2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, IEEE, Jul. 2024, pp. 1–8. doi: 10.1109/ICEEICT61591.2024.10718600.
- [21] S. Matzka, “Predictive Maintenance Dataset (AI4I 2020),” *kaggle*, 2023.
- [22] U. Korat and A. Alimohammad, “Efficient Hardware Implementation of Eigen Solver,” in *2026 IEEE 16th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2026, pp. 1376–1385. doi: 10.1109/CCWC67433.2026.11393888.
- [23] D. K. Yadav, A. Kaushik, and N. Yadav, “Predicting machine failures using machine learning and deep learning algorithms,” *Sustain. Manuf. Serv. Econ.*, vol. 3, 2024, doi: 10.1016/j.smse.2024.100029.
- [24] V. S. Deshmukh, P. Dargude, P. Dhokane, A. Dhurve, and M. K. Nalawade, “Artificial Intelligence And Machine Learning- Based Predictive Maintenance Of Industrial Machines,” vol. 3, no. 5, 2026.
- [25] S. Gupta, Shubham, S. Joshi, and A. Madan, “Optimizing Machine Performance and Reliability: A Predictive Maintenance Approach,” vol. 8, no. 5, pp. 611–614, 2023.
- [26] M. Diwakar, S. Sharma, R. Dhablya, R. Sonar, S. T. Shirkande, and S. Bhattacharya, “AIdriven Strategy for Predicting Equipment Failure in Manufacturing,” in *Proceedings of the 5th International Conference on Information Management & Machine Intelligence*, New York, NY, USA: ACM, Nov. 2023, pp. 1–5. doi: 10.1145/3647444.3647932.