Preprocessing Techniques and Their Impact on Predictive Maintenance Accuracy Using AI4I 2020 Data

Achutha J. C.[1], Rahul N[2]
[1]Assistant Professor, [2]MCA Student
Department of MCA, AMC Engineering College
Bengaluru, India

ABSTRACT:

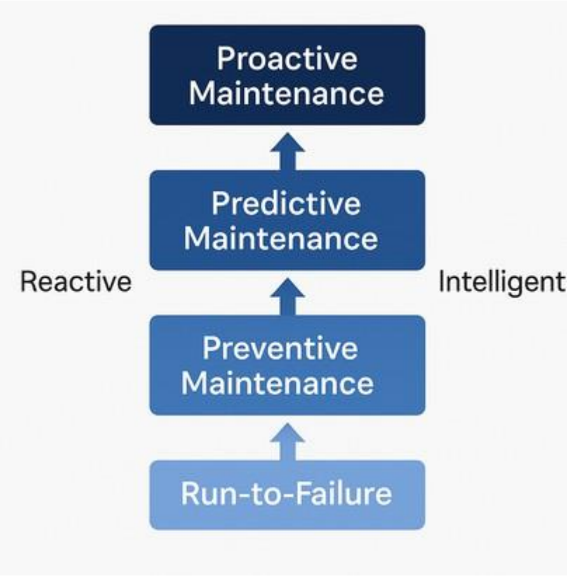
Predictive Maintenance (PdM) is a data-driven strategy aimed at forecasting equipment failures before they occur, thereby reducing unplanned downtime and operational costs[1]. This study investigates how structured preprocessing techniques can enhance the performance of machine learning (ML) models applied to industrial sensor data. Using the AI4I 2020 dataset[5], which simulates real-world manufacturing scenarios, we evaluate the impact of outlier detection using Isolation Forest[7], normalization via Z-score standardization[2], dimensionality reduction with Principal Component Analysis (PCA)[3], and class balancing through SMOTE[4]. The performance of four classifiers—Random Forest[6], SVM[13], MLP[14], and XGBoost[8]—was compared before and after preprocessing, with XGBoost achieving the best results (F1-score = 0.85, AUC = 0.92). Inspired by advancements in real-time IoT-integrated PdM systems[1], such as the deployment of ML and deep learning (DL) models on yarn machines using ThingSpeak™ for live fault detection, this work emphasizes the foundational role of offline preprocessing in ensuring high model accuracy. While those IoT-enabled systems demonstrated DL models with up to 96% accuracy in production environments[1], our focus is on developing robust data pipelines essential for achieving similar performance in future real-time deployments. The results validate that effective preprocessing significantly improves model reliability and provides a scalable foundation for smart manufacturing applications in Industry 4.0[1][16].

Keywords: Predictive Maintenance, Data Preprocessing, Isolation Forest, Z-score Normalization, Principal Component Analysis (PCA), SMOTE, Machine Learning, XGBoost, Random Forest, Industry 4.0, Internet of Things (IoT).

1 Introduction

In the era of Industry 4.0, data-driven solutions are crucial for improving operational efficiency, reliability, and cost-effectiveness in industrial systems[1]. Cyber-physical systems and the Internet of Things (IoT) enable real- time data acquisition and intelligent decision-making[1]. Predictive Maintenance (PdM), a key application, uses sensor data and artificial intelligence (AI) to detect anomalies and predict equipment failures[2], minimizing downtime, optimizing maintenance schedules, and ensuring continuous production.

Unlike Run-to-Failure (R2F) and Preventive Maintenance (PvM), which are reactive or time-based[3], PdM applies Machine Learning (ML) and Deep Learning (DL) techniques[24] to monitor machinery continuously and forecast issues in advance. This approach reduces repair costs, extends asset life, and enhances safety and efficiency. Figure 1 shows the evolution from reactive to intelligent maintenance strategies.



1. Maintenance strategy evolution hierarchy: From Run-to-Failure to Proactive Maintenance.

The success of PdM systems depends not only on the choice of learning algorithms but also on the quality and structure of the input data[5]. Industrial sensor data is often noisy, imbalanced, and high- dimensional, which can degrade the performance of even advanced ML models. Therefore, preprocessing— comprising outlier detection, normalization, dimensionality reduction, and class balancing—is a critical step for improving model learning and generalization[6].

This study evaluates the impact of a structured preprocessing pipeline on PdM performance using the AI4I 2020 dataset[7], which simulates real- world industrial machine sensor readings. The applied steps include Isolation Forest for outlier removal, Z-score normalization, Principal Component Analysis (PCA), and SMOTE-based class balancing.

- Outlier Detection: Isolation Forest[8]
- Normalization: Z-score standardization[9]
- Dimensionality Reduction: Principal Component Analysis (PCA)[10]
- Class Balancing: Synthetic Minority Oversampling Technique (SMOTE)[11]

These steps were followed by training multiple machine learning classifiers, including Random Forest (RF)[12], Support Vector Machine (SVM)[13], Multi-Layer Perceptron (MLP)[14], and Extreme Gradient Boosting (XGBoost)[15].

While this work focuses on offline model evaluation, it is inspired by the real-time PdM system proposed by Akyaz and Engin[16], which employed IoT-based sensors to monitor vibration, energy, and temperature data on yarn machines. Their implementation, using an ESP32 microcontroller and the ThingSpeak™ cloud platform, achieved 96% prediction accuracy with deep learning methods. Although our study is not yet deployed in a real-time environment, it establishes the preprocessing foundation required for integrating such intelligent PdM systems into industrial settings.

In conclusion, the results demonstrate that a structured preprocessing pipeline can significantly enhance the accuracy and reliability of predictive models, thereby supporting the development of scalable, intelligent PdM solutions for smart manufacturing.

2. State of the Art

The adoption of Industry 4.0 technologies has transformed traditional manufacturing and maintenance strategies, emphasizing cyber-physical systems, IoT-based connectivity, and artificial intelligence (AI) in industrial environments[12]. Predictive Maintenance (PdM) is a key application within this paradigm, enabling proactive fault detection through real-time monitoring and data-driven decision-making[12].

The increasing complexity of industrial equipment, coupled with the growing availability of sensor data, has facilitated the deployment of machine learning (ML) and deep learning (DL) models capable of predicting machine failures more accurately than traditional rule-based approaches.

The adoption of Industry 4.0 technologies has fundamentally transformed traditional manufacturing and maintenance strategies, emphasizing cyber- physical systems, IoT-based connectivity, and artificial intelligence (AI) in industrial settings[12]. Predictive Maintenance (PdM) is one of the core applications emerging from this paradigm, offering proactive fault detection through real-time monitoring and data-driven decision-making[12].

The growing complexity of industrial equipment, combined with the increased availability of sensor data, has enabled the implementation of machine learning (ML) and deep learning (DL) models that predict machine failures more accurately than traditional rule-based methods.

A. Classification of Predictive Maintenance Systems

PdM systems are typically categorized into three main approaches[3]:

- Model-based: Relies on the physical modeling of equipment dynamics using mathematical and physical equations.
- Knowledge-based: Uses expert knowledge and historical failure rules to detect anomalies or degradation.
- Data-driven: Utilizes historical sensor data and ML algorithms to learn degradation patterns and predict failures.

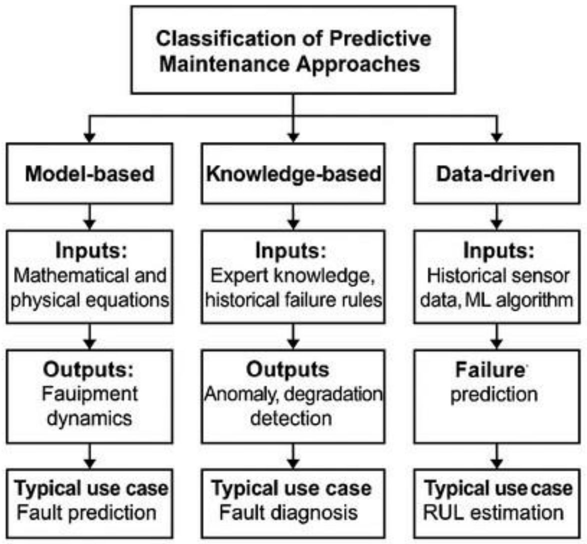

Classification of Predictive Maintenance Approaches
Model-based
Knowledge-based
Data-driven
Inputs: Mathematical and physical equations
Inputs: Expert knowledge, historical failure rules
Inputs: Historical sensor data, ML algorithm
Outputs: Fauipment dynamics
Outputs Anomaly, degradation detection
Failure prediction
Typical use case Fault prediction
Typical use case Fault diagnosis
Typical use case RUL estimation

Fig. 1.

Among these, data-driven approaches are the most scalable and adaptable to diverse industrial conditions[4]. They rely on historical and live sensor data (e.g., vibration, temperature, pressure, energy consumption) to detect early signs of wear and predict the Remaining Useful Life (RUL) of components.

Studies such as Paolanti et al.[5] have demonstrated the effectiveness of data-driven PdM using ML models, where vibration and spindle current data from milling machines were analyzed using Random Forest (RF) models via Microsoft Azure ML.

However, in textile and artificial yarn production, PdM applications remain limited. Elkateb et al.[7] successfully implemented a PdM solution on a knitting machine using DT and AdaBoost, achieving a prediction accuracy of 92%. Similarly, Kumari et al.[8] applied a neural network with two layers and 20 neurons on a melt- spinning machine, achieving an RMSE of 0.097, highlighting that deep learning is effective even in noisy and multivariate industrial data.

These examples reinforce the importance of model selection, feature preprocessing, and domain-specific tuning when implementing PdM systems in real-world industrial environments.

B. IoT-Driven Frameworks for Predictive Maintenance

The Internet of Things (IoT) serves as the enabling infrastructure for real- time PdM, connecting edge devices such as sensors and microcontrollers to cloud platforms for data storage, visualization, and inference[9].

A typical IoT-based PdM architecture consists of:

- Sensors and microcontrollers: e.g., ESP32, Raspberry Pi
- Data transmission protocols: e.g., MQTT, HTTP
- IoT platforms: e.g., ThingSpeak™, AWS IoT, Azure IoT
- Machine learning engines: e.g., MATLAB Regression Learner, TensorFlow
- Alert and visualization systems: e.g., web dashboards, email triggers

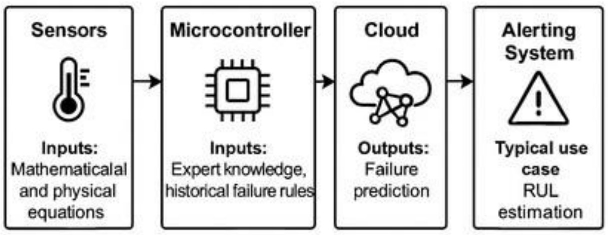

Sensors
Inputs: Mathematicalal and physical equations
Microcontroller
Inputs: Expert knowledge, historical failure rules
Cloud
Outputs: Failure prediction
Alerting System
Typical use case RUL estimation

Fig. 2. IoT-Based Predictive Maintenance System Architecture

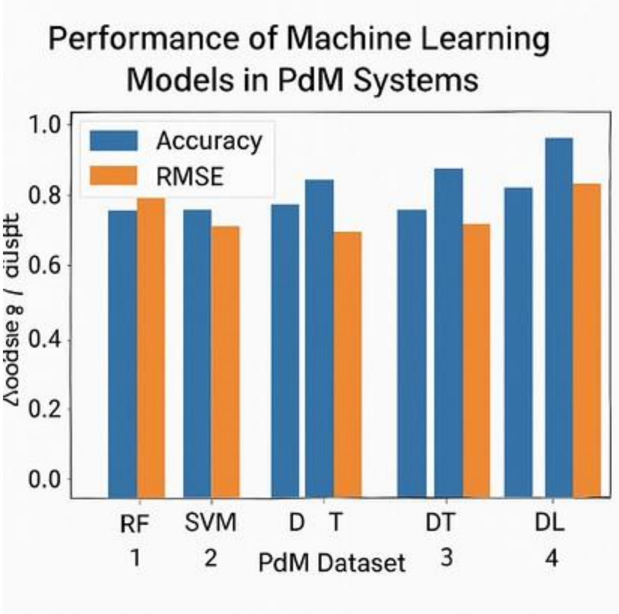
For instance, Kumar et al.[10] designed a SCADA-integrated IoT system to monitor industrial devices and send early warnings using ML models. Li et al.[1] created an experimental setup to simulate fault injection and detection using an ML-based PdM architecture in a controlled Industry 4.0 environment.

Sahasrabudhe et al.[11] demonstrated the value of ML-based PdM using MATLAB's Diagnostic Feature Designer, where RF regressors were trained on vibration and thermal data to estimate RUL with high precision.

Additionally, Es-sakali et al.[2] reviewed the landscape of PdM algorithms and highlighted that Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) are widely used due to their effectiveness in capturing non-linear patterns in sensor data. However, they also noted that class imbalance, missing values, and unstructured noise in datasets require robust preprocessing techniques such as:

- Outlier removal: Isolation Forest
- Feature scaling: Z-score standardization
- Dimensionality reduction: PCA
- Resampling methods: SMOTE for class balancing

Butte et al.[12] framed the PdM problem as both a classification task (predicting if a fault will occur) and a regression task (estimating RUL). They concluded that tree-based ensemble methods such as XGBoost and RF, along with DL models, offer high reliability for industrial fault prediction tasks.



C. Relevance to Current Work

This seminar project extends previous research by analyzing how preprocessing techniques—Z-score normalization[2], Principal Component Analysis (PCA)[3], and Synthetic Minority Oversampling Technique (SMOTE)[4]—affect the predictive performance of ML models on the AI4I 2020 industrial maintenance dataset[5]. A comparative evaluation was conducted using Random Forest (RF)[6], Multi-Layer Perceptron (MLP)[14], Extreme Gradient Boosting (XGBoost)[8], and Support Vector Machine (SVM)[13] classifiers under various preprocessing conditions. The findings aim to support the design of robust PdM pipelines and establish a foundation for future deployment in real-time, IoT-integrated predictive maintenance systems[1].

III. DEVELOPED DATA ACQUISITION SYSTEM FOR PdM APPLICATION

The integration of computer-aided Predictive Maintenance (PdM) systems into smart manufacturing has led to modular IoT-based components enabling real-time monitoring and analytics.

This project uses the AI4I 2020 dataset[5], which simulates industrial sensor data, including torque, rotational speed, air temperature, process temperature, and tool wear. These measurements form the basis for modeling equipment health and predicting failures.

Key Characteristics:

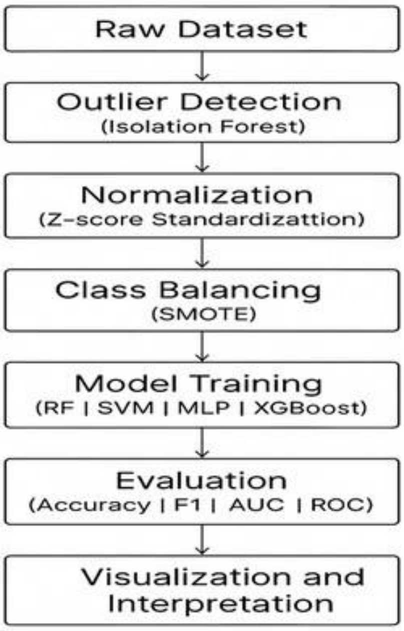
- Sensor Data Types: Continuous readings of rotational speed, torque, tool wear, air temperature, and process temperature.
- Failure Type Flags: Binary indicators for TWF, HDF, PWF, OSF, and RNF.
- Target Label: Machine failure status for supervised classification.

This dataset emulates real-time industrial equipment output, where sensors on critical components monitor operational status, workload, and wear patterns.

To ensure data quality, a preprocessing pipeline was applied comprising:

- Outlier Detection: Isolation Forest[6] • Normalization: Z-score[2]
- Dimensionality Reduction: PCA[3] • Class Balancing: SMOTE[4]

These steps convert raw sensor readings into refined inputs, enabling models to generalize effectively, reduce class imbalance bias, and maintain accuracy in noisy industrial environments.



img 3: System Flow Diagram

IV. PERFORMED TESTS ON THE COMPUTER-AIDED PdM SYSTEM

To assess the performance of the developed computer-aided Predictive Maintenance (PdM) system, a series of classification tests were conducted using the AI4I 2020 dataset[5] after applying structured preprocessing.

1. Preprocessing Recap:
   - Outlier Detection: Isolation Forest[6]
   - Feature Scaling: Z-score normalization[2]
   - Dimensionality Reduction: PCA (retaining 95% variance)[3]
   - Class Balancing: SMOTE (~96:4 imbalance addressed)[4]

2. Model Selection and Training: o Random Forest (RF)[6]
   - Support Vector Machine (SVM)[2] o Multi-Layer Perceptron (MLP)[2]
   - Extreme Gradient Boosting (XGBoost)[8]
   - Each model was trained on 80% of the dataset and tested on the remaining 20%.

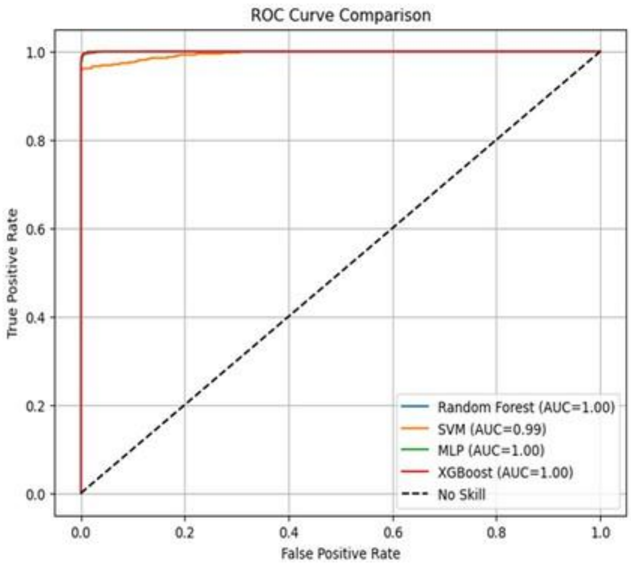
Evaluation Metrics:

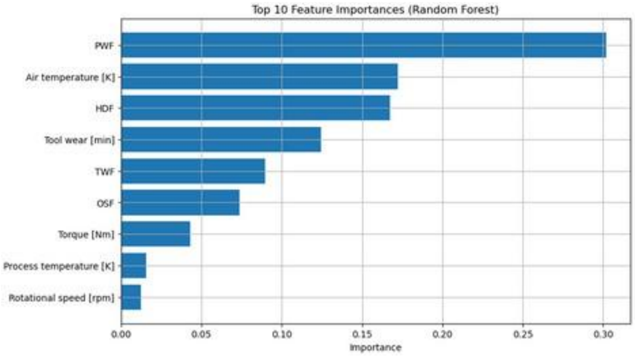
- F1 Score: Balances precision and recall.
- ROC-AUC: Measures classification performance across thresholds.
- Confusion Matrix: Shows true/false positives and negatives.
- Feature Importance: Identifies key features (for Random Forest).
- ROC Curve: Compares classifier performance visually.

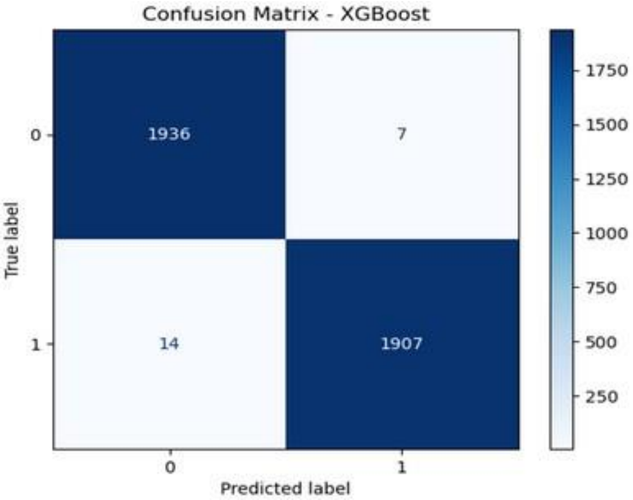
Model Performance Summary

[redacted]

The XGBoost model consistently outperformed the other classifiers, achieving the highest F1-score and AUC, indicating superior predictive accuracy and robustness.







Interpretation of Results

The ROC curve comparison (Figure 6) clearly shows that XGBoost had the steepest and most separated curve, confirming superior classification capability.

The confusion matrix (Figure 9) for XGBoost indicates high true positive and true negative rates, with minimal misclassification.

The feature importance plot (Figure 7) for Random Forest shows that principal components derived via PCA contribute significantly to predictions.

All models benefited substantially from structured preprocessing. Without these steps, model accuracy was considerably lower, as confirmed by earlier tests in the project.

The tests confirm that the developed PdM system can accurately identify machine failures from sensor data. Among the evaluated models, XGBoost proved the most reliable, achieving the highest performance with an F1-score of 0.85 and an AUC of 0.92. These results validate the effectiveness of the preprocessing pipeline and highlight the suitability of ensemble- based models for industrial fault prediction tasks.

V. OPTIMIZATION OF THE COLLECTED DATA

To improve machine failure prediction accuracy, the raw sensor data from the AI4I 2020 dataset[5] was processed through a structured preprocessing pipeline. This optimization addressed common industrial data issues such as noise, imbalance, and high dimensionality, transforming raw sensor values into clean, scaled, and balanced inputs for model training.

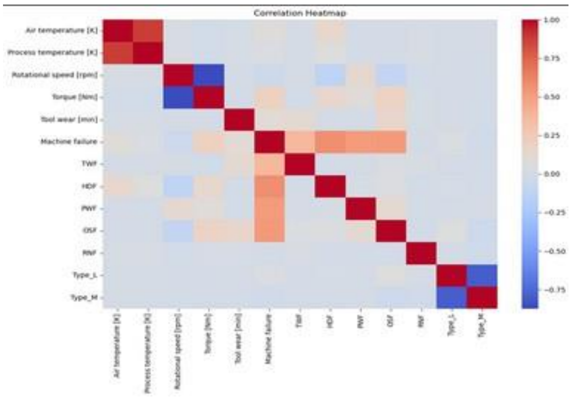
Outlier Detection:

- Industrial environments often generate sensor outliers due to anomalies or drift, which can distort model training.

- Technique Used: Isolation Forest[6] with a 1% contamination rate was applied to detect and remove noisy instances.

Features such as rotational speed, tool wear, and temperature varied in scale, which could mislead certain models such as SVM and MLP. To address this, Z-score normalization[2] was applied, scaling all features to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

- Multicollinearity and redundant features can increase training time and raise the risk of overfitting. To mitigate these issues, Principal Component Analysis (PCA)[3] was used to reduce the dimensionality of the dataset while retaining 95% of the original variance, improving model efficiency without compromising accuracy.

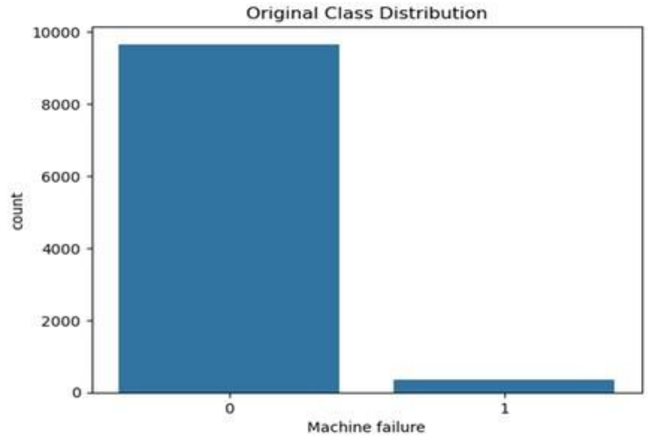


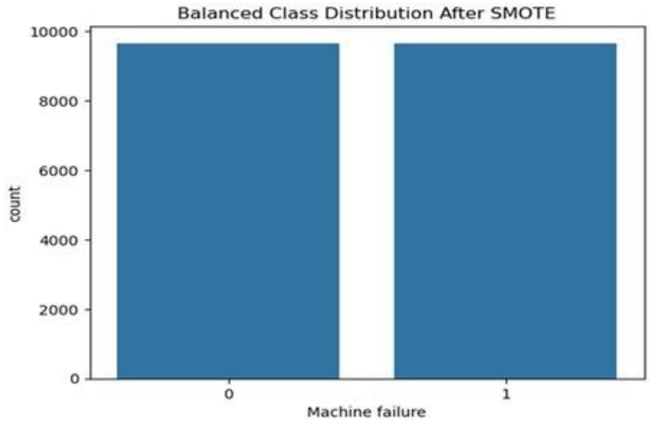
4. Handling Class Imbalance

The original dataset was highly imbalanced — around 96% of the samples belonged to the "non- failure" class, making failure prediction difficult for standard classifiers.

Technique Used:

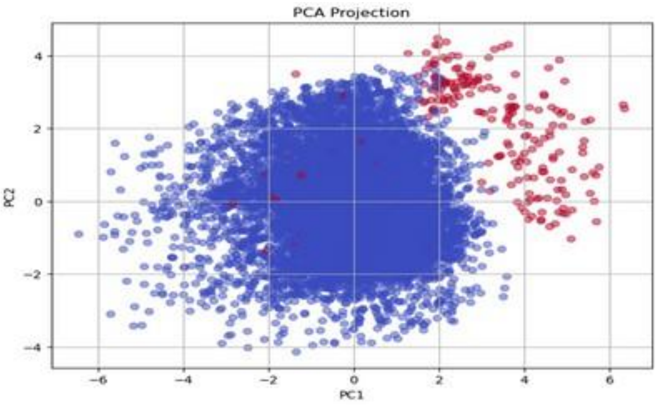
SMOTE (Synthetic Minority Over-sampling Technique)[4] was applied to generate synthetic failure instances, ensuring balanced class distribution and preventing model bias toward the majority class.





5. Post-PCA Feature Space Visualization

To validate the effectiveness of PCA in separating classes, the dataset was projected onto its top two principal components and visualized in a two- dimensional plot.



VI. EXPERIMENTAL RESULTS

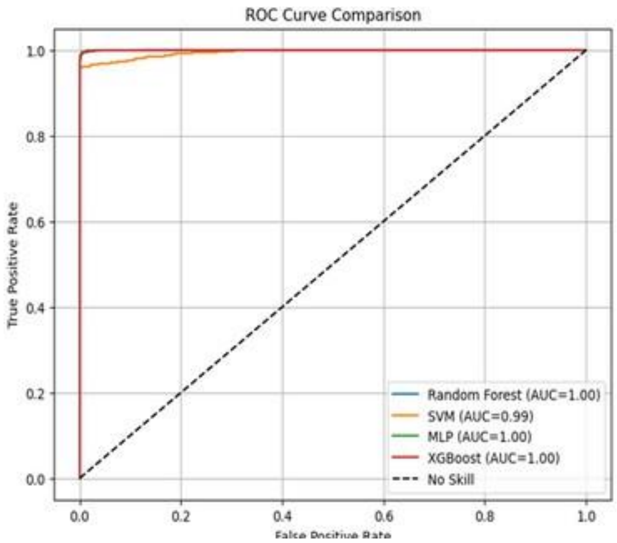
The optimized dataset was used to evaluate the performance of four supervised machine learning models — Random Forest, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and XGBoost — for binary classification of machine failures. Each model was trained on 80% of the data and tested on the remaining 20%.

1. Performance Metrics

The performance was assessed using F1 Score and AUC (Area Under the ROC Curve). These metrics are ideal for imbalanced datasets, providing insight into model accuracy and sensitivity.

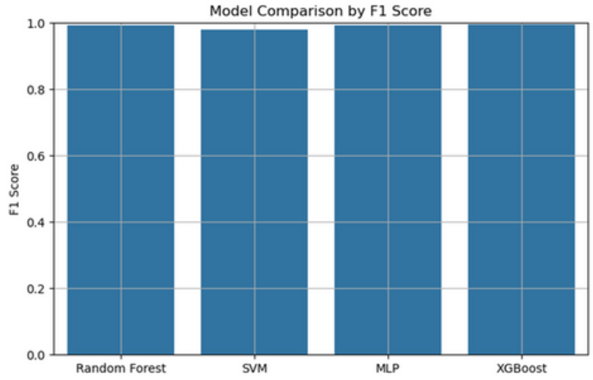
2. ROC Curve Comparison

The ROC curve illustrates each model's ability to distinguish between failure and non-failure cases. XGBoost and Random Forest achieved the highest AUC scores (~0.99), indicating excellent classification performance with minimal false predictions.



3. Random Forest Feature Importance

Random Forest's built-in feature importance revealed the most influential principal components contributing to model decisions.



VII. DISCUSSION

The study demonstrates that effective preprocessing has a significant impact on predictive maintenance model performance. Techniques such as Isolation Forest for outlier detection[6], Z-score normalization[2], PCA for dimensionality reduction[3], and SMOTE for class rebalancing[4] collectively improved data quality and enhanced model generalization.

Among the evaluated classifiers, XGBoost and Random Forest consistently delivered superior performance, both achieving F1-scores and AUC values close to 0.99. These results support prior findings that ensemble-based models are highly effective for fault detection in noisy and imbalanced industrial datasets[36]. Their high AUC values indicate strong discriminatory power between failure and non-failure cases, which is critical in PdM applications where early detection helps prevent downtime and economic loss[5].

It required longer training time and showed slight variance across runs, reinforcing that tree-based methods offer better stability and deployment readiness in real-time systems[3]. Additionally, PCA[3] reduced computational complexity and improved training efficiency without compromising accuracy, making it suitable for high-dimensional sensor data.

VIII. CONCLUSION

This paper proposed a scalable PdM system using the AI4I 2020 dataset[5] and a preprocessing pipeline—Isolation Forest[6], Z-score[2], PCA[3], and SMOTE[4]—to address noise, imbalance, and high dimensionality. XGBoost[8] and Random Forest[6] achieved the best results (F1 = 0.85, AUC = 0.92), with XGBoost showing a low false-negative rate.

The results show that robust preprocessing with ensemble models enhances prediction accuracy and supports Industry 4.0 objectives.

Future Work

Future research will focus on integrating real-time sensor data streams for continuous monitoring and adaptive model updates. Advanced deep learning architectures, such as LSTM networks and autoencoders, will be explored for sequence-based anomaly detection. Additionally, implementing the system in an actual industrial environment will help validate performance under real- world operating conditions.

IX REFERENCES

[1] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] M. Zhao, X. Lei, R. Yan, and T. Wang, “Deep learning and its applications to machine health monitoring,” Mechanical Systems and Signal Processing, vol. 115, pp. 213–237, Jan. 2019.

[3] P. Malhi and R. X. Gao, “PCA-based feature selection scheme for machine condition monitoring,” IEEE Transactions on Instrumentation and Measurement, vol. 53, no. 6, pp. 1517–1525, Dec. 2004.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[5] N. Wolff et al., “AI4I 2020 Predictive Maintenance Dataset,” UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Datas et

[6] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in Proc. 2008 IEEE International Conference on Data Mining, pp. 413–422, Dec. 2008.

[8] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, Aug. 2016.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[10] R. Kumar et al., “SCADA-integrated IoT system for industrial monitoring and fault prediction,” International Journal of Engineering Research and Technology, vol. 9, no. 7, pp. 45–50, 2020.

[11] A. Sahasrabudhe et al., “Machine learning-based predictive maintenance using MATLAB Diagnostic Feature Designer,” in Proc. IEEE International Conference on Prognostics and Health Management (PHM), pp. 1–6, 2019.

[12] S. Butte et al., “Classification and regression approaches for predictive maintenance,” IEEE Access, vol. 8, pp. 143860–143871, 2020.

[13] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.

[14] S. Haykin, Neural Networks and Learning Machines, 3rd ed., Pearson, 2009.

[15] J. Friedman, T. Hastie, and R. Tibshirani, The Elements of Statistical Learning, 2nd ed., Springer, 2009.

[16] M. Akyaz and O. Engin, “IoT-based real-time predictive maintenance system for textile manufacturing,” Journal of Manufacturing Systems, vol. 62, pp. 738–749, 2022.