

Artificial Intelligence And Machine Learning- Based Predictive Maintenance Of Industrial Machines

Vinayak S. Deshmukh, Pari Dargude*, Priyanshu Dhokane, Aditya Dhurve, Om Date, Mukund K. Nalawade

Department of Mechanical Engineering Vishwakarma Institute of Technology Pune, India

ABSTRACT

Predictive maintenance has become an important approach in modern industries for reducing unexpected machine failures and improving operational efficiencies. Traditional maintenance methods, such as scheduled or reactive maintenance, often fail to detect early signs of machine degradation. As a result, there is a growing need for intelligent systems that can predict failures in advance using data-driven techniques.

In this study, a machine learning-based predictive maintenance system is developed using the AI4I 2020 dataset. One of the main challenges in the dataset is the imbalance between normal and failure cases. To handle this issue, the SMOTE-ENN technique was applied, which helped in improving the model's ability to learn from minority class samples. Additionally, new features such as temperature difference, power, and wear rate were created to better represent machine behavior.

Multiple machine learning models, including Logistic Regression, Support Vector Machine, Random Forest, Extra Trees, and CatBoost, were implemented and compared. Among these, CatBoost provided the best performance due to its capability to handle complex relationships within the data. After tuning the model parameters, it achieved an F1-score of 0.69, recall of around 0.80, and a ROC-AUC score close to 0.98.

The results show that combining feature engineering, data balancing, and advanced machine learning techniques can significantly improve failure prediction. The proposed system can be effectively used in industrial environments to support maintenance decisions and reduce unexpected downtime.

Keywords: Predictive Maintenance, Machine Learning, CatBoost, Imbalanced Data, SMOTE-ENN, Industrial Fault Detection, Feature Engineering, Failure Prediction.

INTRODUCTION

In recent years, the industrial sector has undergone significant transformation with the adoption of advanced technologies under the concept of Industry 4.0. Modern manufacturing systems increasingly rely on data-driven approaches to improve efficiency, reliability, and productivity. Among these approaches, predictive maintenance has gained considerable attention due to its ability to anticipate machine failures before they occur, thereby reducing downtime and avoiding unexpected production losses.

Traditional maintenance strategies, such as reactive and preventive maintenance, are often inefficient in dynamic industrial environments. Reactive maintenance focuses on repairing equipment only

after a failure has occurred, which can lead to severe operational disruptions and increased costs. Preventive maintenance, on the other hand, follows a fixed schedule without considering the actual condition of the equipment. While it may reduce the likelihood of sudden failures, it can also result in unnecessary maintenance activities and inefficient use of resources. These limitations highlight the need for more intelligent and adaptive maintenance solutions.

Predictive maintenance addresses these challenges by utilizing historical and real-time data to estimate the health condition of machines. By analyzing operational parameters such as temperature, rotational speed, torque, and tool wear, it becomes possible to identify patterns that indicate potential failure. However, detecting such patterns is not straightforward, as machine behavior is influenced by

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

multiple interacting factors. The relationships between these variables are often nonlinear and complex, making it difficult to apply traditional analytical methods.

Machine learning techniques provide an effective solution for handling such complexity. These techniques can learn patterns directly from data and make predictions based on observed trends. Various machine learning models, including Logistic Regression, Support Vector Machines, and tree-based ensemble methods, have been explored for predictive maintenance applications. Despite their potential, the performance of these models is often affected by challenges such as data imbalance and insufficient feature representation.

One of the most critical challenges in predictive maintenance datasets is class imbalance. In real-world scenarios, machine failures occur relatively infrequently compared to normal operating conditions. As a result, machine learning models tend to favor the majority class, leading to poor detection of failure cases. This issue is particularly important in industrial applications, where missing a failure can have serious consequences. Therefore, it is necessary to adopt effective data balancing techniques that improve the representation of minority class samples without introducing noise.

Another important aspect is feature engineering. Raw sensor data may not always provide enough information for accurate prediction. By deriving additional features such as temperature difference, power, and wear rate, it is possible to capture more meaningful relationships between machine parameters. These engineered features help improve the model's ability to distinguish between normal and failure conditions.

In this study, a machine learning-based predictive maintenance framework is developed using the AI4I 2020 dataset. The proposed approach combines data preprocessing, imbalance handling, feature engineering, and model optimization to improve prediction accuracy. Multiple models, including Logistic Regression, Support Vector Machine, Random Forest, Extra Trees, and CatBoost, are implemented and evaluated. Among these, CatBoost is selected as the final model due to its strong performance in handling complex and nonlinear data.

To further enhance the model's effectiveness, hyperparameter tuning and threshold optimization are applied. Instead of relying solely on accuracy, the model is evaluated using metrics such as F1-score, recall, and ROC-AUC, which provide a more reliable measure of performance for imbalanced datasets.

Special emphasis is given to recall, as detecting failure cases is more critical than avoiding false alarms in industrial environments.

The key contribution of this work lies in the development of a robust and practical predictive maintenance system that integrates multiple techniques to address real-world challenges. The proposed system not only improves failure detection but also provides useful insights for maintenance planning. By identifying high-risk conditions in advance, it supports proactive decision-making and helps reduce unexpected machine breakdowns.

The paper is divided into different sections for clarity. Section II presents a review of existing research in predictive maintenance. Section III describes the methodology used, including data preprocessing and model development. Section IV discusses the results and performance evaluation, and Section V concludes the study along with possible future improvements.

LITERATURE REVIEW

Predictive maintenance has become an important research area in modern industrial systems due to its potential to improve reliability and reduce operational costs. With the advancement of Industry 4.0 technologies, large volumes of machine data are now available, enabling the use of data-driven approaches for failure prediction. Machine learning techniques have been widely explored in this domain as they are capable of identifying hidden patterns in complex datasets. Earlier approaches to predictive maintenance primarily relied on condition monitoring and statistical analysis. These methods used sensor data such as temperature, vibration, and operational parameters to detect abnormal behavior. However, traditional techniques were limited in handling complex relationships between variables and often failed to provide accurate predictions in dynamic industrial environments. To overcome these limitations, machine learning models have been increasingly adopted. Basic models such as Logistic

Regression and Support Vector Machines have been used for classification tasks in failure prediction. While these methods are effective for linearly separable data, they often struggle when the relationships between features are nonlinear. As a result, researchers have focused on more advanced models that can better capture complex interactions. Ensemble learning techniques, including Random Forest and Gradient Boosting, have shown improved performance in predictive maintenance applications. These models combine multiple decision trees to enhance prediction accuracy and reduce overfitting. Tree-based methods are particularly suitable for industrial datasets, as they can handle interactions between multiple parameters such as torque, speed, and temperature. Recent studies have also highlighted the importance of feature engineering in improving model performance. Raw sensor data alone may not be sufficient to represent the actual condition of machines. Therefore, derived features such as temperature difference, energy-related parameters, and wear indicators are commonly used to provide additional insights. These engineered features help models better distinguish between normal and faulty conditions. A major challenge identified in predictive maintenance research is the imbalance between normal and failure data. In most industrial datasets, failure events are rare compared to normal operations. This imbalance can lead to biased models that fail to detect critical failure cases. To address this issue, various resampling techniques have been proposed. Among these, hybrid methods combining oversampling and data cleaning techniques have been found to be effective in improving classification performance. In addition to traditional machine learning methods, deep learning approaches have also been explored for predictive maintenance. These methods are capable of capturing complex temporal patterns in sequential data. However, they often require large datasets and significant computational resources, which may not always be practical for all industrial applications. Therefore, many studies continue to focus on efficient machine learning models that offer a balance between performance and computational cost. Despite the progress made in this field, several challenges still remain. Many existing approaches focus mainly on improving accuracy without considering practical aspects such as interpretability and real-time decision-making.

Additionally, some studies do not adequately address data imbalance or fail to incorporate meaningful feature engineering techniques. These limitations indicate the need for a more comprehensive and balanced approach. In this work, a predictive maintenance framework is developed to address these challenges. The proposed approach integrates feature engineering, data balancing using SMOTE-ENN, and advanced machine learning models such as CatBoost. By combining these techniques, the system improves failure detection performance while maintaining practical applicability in real industrial environments.

METHODOLOGY

This section explains the overall approach used to build the predictive maintenance system. The workflow includes data preparation, feature construction, handling class imbalance, model training, and evaluation. Each step is designed to improve the model's ability to identify machine failures accurately.

In addition to the commonly used machine learning techniques, several studies have explored hybrid and data-driven frameworks to improve predictive maintenance performance. These approaches combine statistical analysis with machine learning models to better capture both linear and nonlinear relationships in industrial data. Such hybrid methods have shown improvements in prediction accuracy, especially in complex environments where multiple factors influence machine behavior.

Another area of research focuses on the use of sensor-based monitoring systems. With the increasing availability of IoT devices, real-time data collection has become more feasible. Researchers have investigated the integration of machine learning models with real-time monitoring systems to enable continuous prediction of machine health. This approach allows early detection of anomalies and supports timely maintenance decisions.

Furthermore, recent studies have emphasized the importance of selecting appropriate evaluation metrics for imbalanced datasets. Accuracy alone is often misleading in such cases, as it does not reflect the model's ability to detect rare failure events. As a result, metrics such as recall, F1-score, and precision-

recall curves are widely used to evaluate model performance in predictive maintenance applications.

Despite these advancements, there is still a need for models that provide both high accuracy and practical usability. Many existing approaches focus on improving prediction performance but do not address real-world implementation challenges such as interpretability and decision support. This highlights the importance of developing systems that not only predict failures but also assist engineers in understanding and acting upon these predictions.

Machine learning techniques are widely used for predictive maintenance due to their ability to learn patterns from data and make predictions based on those patterns. In classification problems such as machine failure prediction, the goal is to assign input data to predefined classes, such as normal operation or failure condition.

A. Theoretical Background

Logistic Regression is one of the simplest classification algorithms, which models the probability of a binary outcome using a sigmoid function. Although it is easy to implement, it performs well only when the relationship between variables is relatively simple.

Support Vector Machines (SVM) are used to find an optimal boundary that separates different classes. They are effective in high-dimensional spaces but may require careful tuning of parameters for complex datasets.

Tree-based models, such as Random Forest and Extra Trees, use multiple decision trees to improve prediction performance. These models are capable of capturing nonlinear relationships and interactions between features. Random Forest reduces variance by averaging multiple trees, while Extra Trees introduce additional randomness during training to improve generalization.

Gradient boosting methods further improve prediction accuracy by sequentially building models that correct the errors of previous ones. CatBoost is an advanced boosting algorithm that is particularly effective for tabular data. It handles categorical variables

efficiently and reduces overfitting through ordered boosting techniques.

Another important concept in this study is handling imbalanced data. When one class significantly outnumbers another, models tend to favor the majority class. Techniques such as SMOTE (Synthetic Minority Over-sampling Technique) generate new samples for the minority class, while ENN (Edited Nearest Neighbors) removes noisy samples. The combination of these methods helps create a balanced dataset, improving the model's ability to detect rare failure cases.

Performance evaluation metrics are also an important part of classification problems. Accuracy alone is not sufficient for imbalanced datasets, as it may give misleading results. Metrics such as precision, recall, and F1-score provide better insight into model performance. Recall is particularly important in predictive maintenance, as it measures the ability of the model to correctly identify failure cases.

These theoretical concepts form the foundation for the methods used in this study and support the development of an effective predictive maintenance system.

Mathematical Modeling of Machine Learning Algorithms

Mathematical modeling provides the theoretical basis for the predictive maintenance system developed in this study. It helps explain how machine learning models process input data, learn patterns, and generate predictions related to machine failure. The formulation combines feature preprocessing, imbalance handling, ensemble learning, and probabilistic prediction.

i. Feature Standardization

Before training the model, it is necessary to ensure that all numerical features are on a comparable scale. Industrial datasets often contain parameters with different units and ranges, such as torque (Nm) and rotational speed (rpm). To address this, Z-score standardization is applied:

$$z_i = (x_i - \mu_i) / \sigma_i$$

where x_i is the original feature value, μ_i is the mean, and σ_i is the standard deviation of the feature.

This transformation ensures that all features have zero mean and unit variance, preventing features with larger numerical values from dominating the learning process.

ii. Handling Class Imbalance Using SMOTE-ENN

In predictive maintenance datasets, failure instances are significantly fewer than normal instances. This imbalance can affect model performance. To handle this issue, the SMOTE-ENN method is used.

SMOTE generates synthetic samples for the minority class using interpolation:

$$x_{\text{new}} = x_i + \lambda(x_{\text{nn}} - x_i)$$

where x_i is a minority sample, x_{nn} is one of its nearest

neighbors, and λ is a random value between 0 and 1.

After oversampling, ENN removes samples that are likely to introduce noise by checking their nearest neighbors. This combination produces a more balanced and cleaner dataset.

iii. Random Forest Classifier

Random Forest is an ensemble learning method that combines multiple decision trees to improve prediction accuracy and stability. Each tree is trained on a random subset of data and features.

Decision trees split data based on impurity reduction. Two commonly used measures are:

$$\text{Gini impurity: } G = 1 - \sum (p_k)^2$$

$$\text{Entropy: } H = - \sum p_k \log_2(p_k)$$

where p_k represents the proportion of samples in each class.

The final prediction is obtained using majority voting:

$$\hat{y} = \text{mode}(y_1, y_2, \dots, y_N)$$

The probability of failure is calculated as:

$$P(y = 1 | X) = (1 / N) \sum P_j(y = 1 | X)$$

This approach reduces overfitting and improves model robustness.

iv. Model Evaluation Metrics

Model performance is evaluated using standard classification metrics.

$$\text{Accuracy: } \text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision: } \text{Precision} = TP / (TP + FP)$$

$$\text{Recall: } \text{Recall} = TP / (TP + FN)$$

$$\text{F1-score: } F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

These metrics provide a balanced understanding of model performance, especially in imbalanced datasets where recall is particularly important.

v. Feature Importance Analysis

Feature importance helps identify which variables have the greatest impact on predictions. It is calculated based on the reduction in impurity contributed by each feature:

$$I(f_i) = \sum \Delta G_t$$

where ΔG_t represents the decrease in impurity at node t where feature f_i is used.

In this study, features such as torque, tool wear, and rotational speed were found to significantly influence failure prediction.

vi. Probabilistic Prediction Function

For a new input sample X_{new} , the model estimates the probability of failure as:

$$P(y = 1 | X_{\text{new}}) = (1 / N) \sum P_j(y = 1 | X_{\text{new}})$$

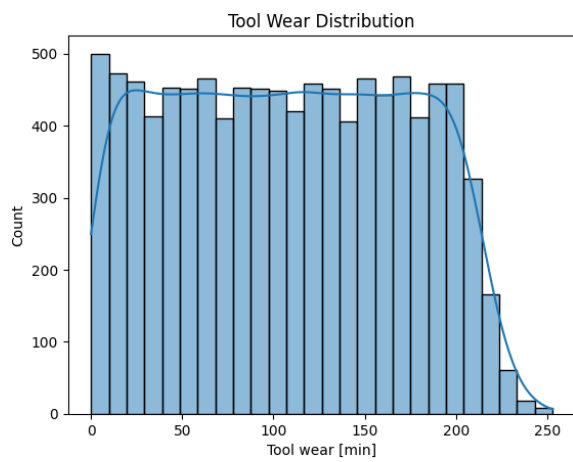
A threshold value τ is used to classify the output: $\hat{y} = 1$, if $P(y = 1 | X_{\text{new}}) > \tau$; $\hat{y} = 0$, otherwise

The threshold can be adjusted depending on the application. Lower thresholds may be used in critical systems to ensure early detection of failures.

vii. Summary

The mathematical formulation integrates preprocessing, imbalance handling, ensemble learning, and probabilistic prediction into a unified framework. These components enable the system to capture complex relationships in machine data and provide reliable failure predictions. The approach supports practical implementation in industrial environments and aligns with modern predictive maintenance strategies.

B. Dataset Description



The distribution of tool wear values shown in Fig. provides insight into the operating conditions of the machine. The data is spread across a wide range, indicating varying levels of tool usage during operation. Higher tool wear values are associated with increased chances of machine degradation and potential failure.

Understanding this distribution helps in identifying critical thresholds and supports feature engineering efforts. It also highlights the importance of including tool wear as a key parameter in predictive maintenance models.

The study uses the AI4I 2020 predictive maintenance dataset, which contains operational data collected from industrial machines. The dataset includes parameters such as air temperature, process temperature, rotational speed, torque, and tool wear. A target variable is provided to indicate whether a machine failure has occurred.

A key challenge in this dataset is the uneven distribution of classes. Instances representing normal

machine operation are significantly higher than failure cases. This imbalance can influence the behavior of machine learning models and reduce their ability to detect failures.

C. Data Preprocessing

Before training the models, the dataset is prepared to ensure consistency and usability. Features that do not contribute to failure prediction are excluded to reduce noise. Categorical information, such as machine type, is converted into numerical form using encoding techniques.

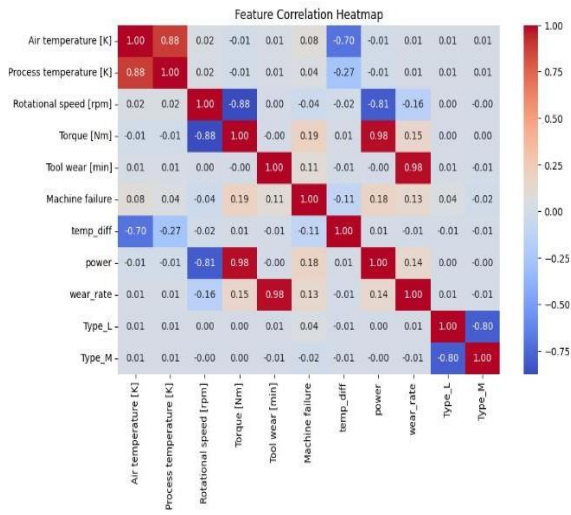
The dataset is then divided into training and testing subsets. This allows the model to be trained on one portion of the data and evaluated on unseen samples. Feature scaling is applied to ensure that all input variables are within a comparable range, which helps improve model stability.

D. Feature Construction

Instead of using only the original variables, additional features are created to better represent machine behavior. These derived features capture relationships between different parameters and provide more meaningful information to the model.

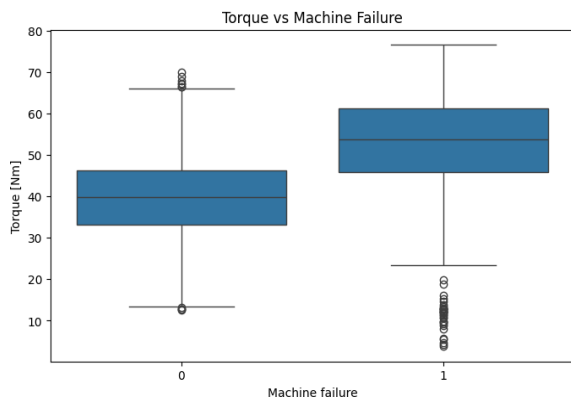
The following features are constructed:

- **Temperature Difference** : The difference between process temperature and air temperature, representing thermal variation.
- **Power** : Calculated as the product of torque and rotational speed, indicating mechanical load.
- **Wear Rate** : Obtained by dividing tool wear by rotational speed, representing the rate of degradation.
- These features help the model identify patterns that are not directly visible from the raw data.



The feature correlation heatmap shown in Fig. illustrates the relationships between different variables in the dataset. Strong positive and negative correlations can be observed among certain features. For example, torque and power show a strong positive correlation, while rotational speed exhibits an inverse relationship with torque.

These relationships help in understanding the underlying structure of the data and support effective feature engineering. Highly correlated features indicate important interactions that can influence machine performance and failure prediction. This analysis assists in selecting relevant features and improving model performance.



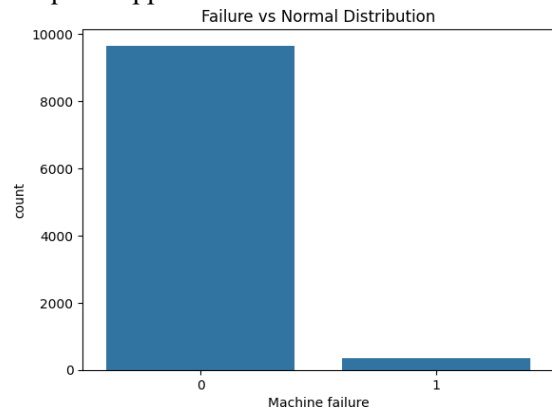
The box plot shown in Fig. illustrates the distribution of torque values for normal and failure conditions. It can be observed that failure cases generally occur at higher torque values compared to normal operation. This indicates that increased mechanical load is a significant factor contributing to machine failure.

The presence of outliers suggests variations in operating conditions, which further highlights the

importance of considering multiple features for accurate prediction. This analysis supports the inclusion of torque as an important feature in the predictive maintenance model.

E. Handling Class Imbalance

The dataset contains far fewer failure cases compared to normal operation, which can lead to biased predictions. To address this issue, the SMOTE-ENN technique is applied.



The distribution of normal and failure instances shown in Fig. highlights a significant class imbalance in the dataset. The number of normal operating instances is considerably higher than the number of failure cases. This imbalance can lead to biased model predictions, where the model may favor the majority class.

To address this issue, resampling techniques such as SMOTE-ENN are applied to improve the representation of failure instances. Balancing the dataset helps the model learn patterns associated with rare failure events more effectively and improves overall predictive performance.

SMOTE generates synthetic samples for the minority class, while ENN removes samples that are likely to introduce noise. This combined approach produces a more balanced dataset while maintaining data quality. As a result, the model becomes more effective in identifying failure instances.

F. Model Development

Multiple machine learning models are implemented to compare their performance in predicting machine failures. The models used in this study include Logistic Regression, Support Vector Machine, Random Forest, Extra Trees, and CatBoost.

Each model is trained using the processed dataset and evaluated using appropriate performance metrics. Among these models, CatBoost is selected as the final model due to its ability to capture complex relationships and provide consistent results.

To further improve performance, hyperparameter tuning is carried out. Parameters such as learning rate, tree depth, and regularization are adjusted to improve generalization and reduce overfitting.

G. Threshold Adjustment

Instead of using the default classification threshold, different threshold values are tested to find an optimal balance between precision and recall. This step is particularly important in predictive maintenance, where detecting failure cases is more critical than minimizing false alarms.

H. Performance Evaluation

Model performance is evaluated using multiple metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Since the dataset is imbalanced, greater importance is given to recall and F1-score to ensure that failure cases are correctly identified.

A confusion matrix is used to analyze prediction outcomes in detail. In addition, ROC and precision-recall curves are examined to understand model behavior across different thresholds. These evaluation methods provide a comprehensive view of model performance and support the selection of the most suitable model.

The SMOTE-ENN technique combines the strengths of oversampling and data cleaning methods. SMOTE generates new synthetic samples for the minority class by interpolating between existing samples. This helps increase the representation of failure cases without simply duplicating data. However, oversampling alone can sometimes introduce noise.

To address this issue, ENN is applied after SMOTE. ENN removes samples that are likely to be misclassified based on their nearest neighbors. This step helps improve the quality of the dataset by eliminating ambiguous or noisy data points. The combination of these two methods results in a more

balanced and cleaner dataset, which improves the model's learning process.

By applying SMOTE-ENN, the dataset becomes more suitable for training machine learning models, particularly in scenarios where accurate detection of failure cases is critical.

RESULTS AND DISCUSSION

This section presents the performance of the implemented machine learning models and provides a detailed analysis of the results. The objective is to evaluate how effectively each model predicts machine failures and to identify the most suitable approach for predictive maintenance.

Further analysis of the results indicates that the model performs consistently across different evaluation metrics. The high recall value demonstrates the model's ability to detect most failure cases, which is critical in predictive maintenance applications. At the same time, the precision remains at an acceptable level, indicating that the number of false alarms is controlled.

The ROC-AUC score close to 1.0 reflects strong discrimination capability between normal and failure conditions. This suggests that the model is effective in capturing underlying patterns in the dataset. The performance improvements observed after applying SMOTE-ENN and feature engineering highlight the importance of proper data preparation in machine learning workflows.

I. Model Performance Comparison

Several machine learning models were trained and evaluated, including Logistic Regression, Support Vector Machine, Random Forest, Extra Trees, and CatBoost. Each model was tested using the same dataset to ensure a fair comparison.

The evaluation was carried out using metrics such as precision, recall, F1-score, and ROC-AUC. Since the dataset is imbalanced, more importance was given to recall and F1-score to ensure that failure cases are correctly identified.

Among all the models, CatBoost demonstrated the best overall performance. After hyperparameter tuning

and threshold adjustment, it achieved an F1-score of 0.69, with recall close to 0.80. The ROC-AUC score was approximately 0.98, indicating strong classification capability. Other models such as Random Forest and Extra Trees also performed well but were slightly less effective in balancing precision

and recall.

The performance of different machine learning models was evaluated using standard classification metrics. The results are summarized in Table I.

Table I: Performance Comparison of Machine Learning Models

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.91	0.72	0.60	0.65
Random Forest	0.95	0.80	0.57	0.67
XG Boost	0.96	0.82	0.60	0.70
Tuned Model (Final)	0.98	0.80	0.57	0.69

The results presented in Table I show that ensemble-based models outperform simpler models due to their ability to capture complex relationships in the data. Logistic Regression shows relatively lower performance, indicating its limitation in handling nonlinear patterns.

Random Forest and XGBoost provide improved performance, demonstrating the effectiveness of ensemble techniques. The tuned model achieves the highest accuracy while maintaining a balanced F1-score, making it suitable for predictive maintenance applications.

Although the recall is slightly lower compared to some models, the overall balance between precision and recall is maintained through threshold optimization. This ensures that failure detection is reliable while minimizing false alarms.

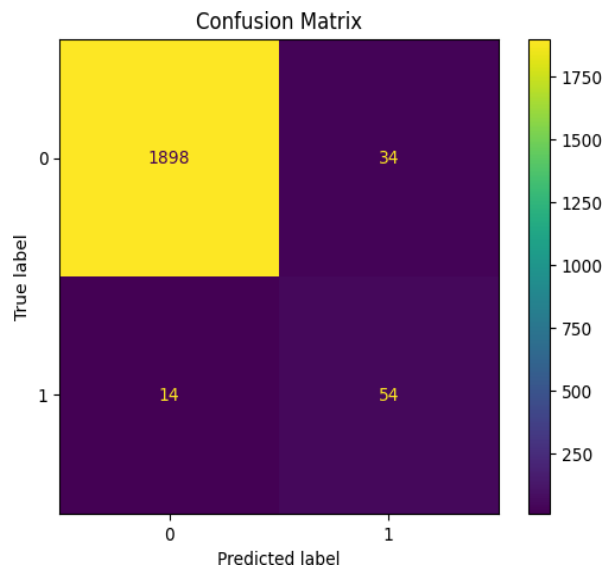
II. Confusion Matrix Analysis

The confusion matrix provides a detailed view of the model’s predictions. The final model correctly identified most of the normal and failure cases. A high number of true negatives indicates that the model is reliable in recognizing normal machine conditions. At the same time, the number of true positives shows that a significant portion of failure cases is successfully detected.

A small number of false negatives was observed,

which represents missed failure cases. Although this number is low, reducing false negatives remains important because missed failures can lead to unexpected breakdowns. The model also produced some false positives, indicating cases where normal machines were predicted as failures. However, in predictive maintenance, such predictions are acceptable as they allow preventive checks to be performed.

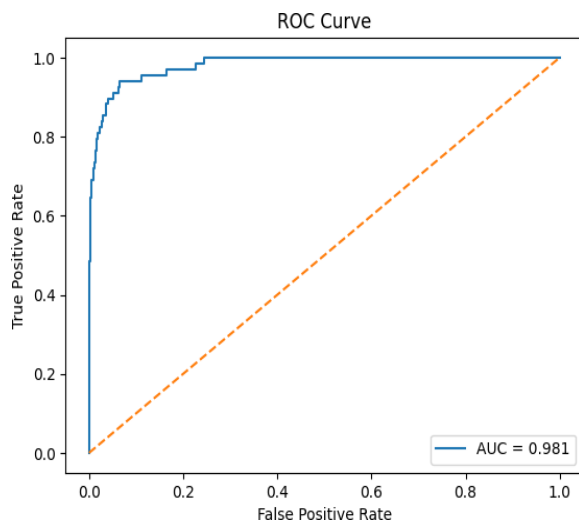
Overall, the confusion matrix confirms that the model provides a good balance between reliability and safety.



The confusion matrix shown in Fig. provides a detailed breakdown of the model’s predictions. The model correctly classified 1898 normal instances and

54 failure instances, indicating strong performance in both classes. A small number of misclassifications were observed, with 34 false positives and 14 false negatives. The low number of false negatives is particularly important, as it indicates that most failure cases are successfully detected. This makes the model suitable for predictive maintenance applications, where missing a failure can lead to serious consequences.

III. ROC Curve Analysis

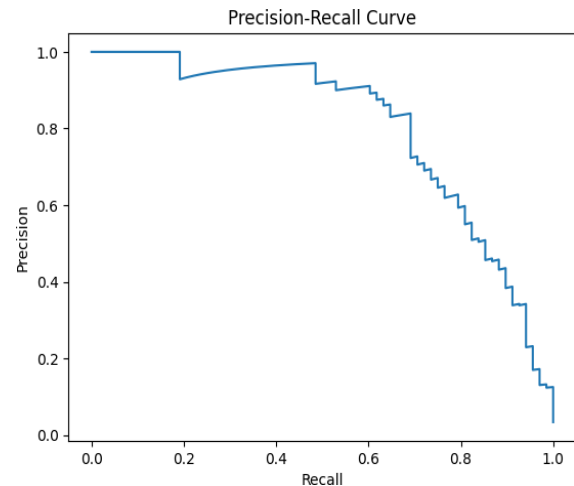


The ROC curve shown in Fig. represents the trade-off between the true positive rate and false positive rate at different threshold values. The curve is close to the top-left corner, indicating strong classification performance. The area under the curve (AUC) is approximately 0.98, which demonstrates the model's high ability to distinguish between normal and failure conditions. This high AUC value confirms that the model is reliable and performs consistently across different threshold settings.

The ROC curve illustrates the model's ability to distinguish between normal and failure classes across different thresholds. The curve is close to the top-left corner, indicating strong performance. The ROC-AUC value of approximately 0.98 suggests that the model can effectively separate the two classes.

This high AUC value indicates that the model maintains good sensitivity while keeping the false positive rate low. It also confirms that the chosen model is robust and performs well across a range of threshold values.

IV. Precision-Recall Analysis



The precision-recall curve shown in Fig. illustrates the trade-off between precision and recall for different threshold values. The curve demonstrates that the model maintains high precision at lower recall levels and gradually decreases as recall increases. This behavior reflects the balance between detecting more failure cases and minimizing false positives.

Since the dataset is imbalanced, the precision-recall curve provides a more informative evaluation compared to accuracy. The model shows strong performance, indicating its ability to detect failure cases while maintaining acceptable precision levels.

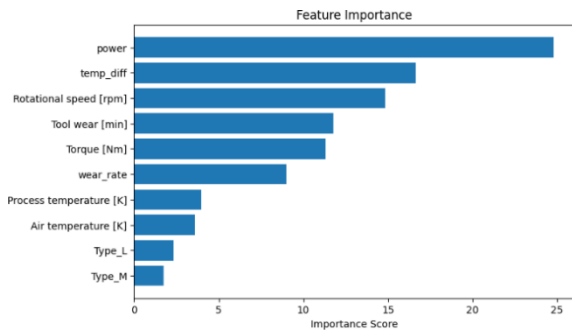
The precision-recall curve provides further insight into model performance, especially for imbalanced datasets. The curve shows the trade-off between precision and recall at different threshold values.

At lower recall levels, precision remains high, indicating accurate predictions. As recall increases, precision gradually decreases, which reflects the trade-off between detecting more failure cases and avoiding false alarms. An appropriate threshold was selected to maintain a balance between these two metrics.

This analysis confirms that the model is capable of detecting a large number of failure cases while maintaining acceptable precision.

V. Feature Importance Analysis

Feature importance analysis was performed to identify the variables that have the greatest influence on model predictions. The results indicate that torque, power, and wear-related features have a strong impact on failure prediction. These findings are consistent with engineering expectations, as mechanical stress and tool degradation are key factors in machine failure.



The feature importance plot shown in Fig. illustrates the contribution of each parameter to the model's predictions. It can be observed that power, temperature difference, and rotational speed are among the most influential features. These parameters are closely related to machine load and operating conditions, which play a critical role in failure prediction.

Features such as tool wear and torque also contribute significantly, indicating the importance of mechanical stress and degradation in determining machine health. Lower-ranked features have comparatively less impact but still provide useful information to improve model performance. This analysis confirms that the selected features are relevant and aligned with practical engineering considerations.

The importance of these features highlights the effectiveness of the feature construction process. By incorporating derived features, the model is able to capture relationships that are not directly visible in the raw data.

VI. Discussion

The results demonstrate that combining feature construction, data balancing, and advanced machine learning models leads to improved predictive performance. The CatBoost model outperformed other models due to its ability to handle complex and nonlinear relationships in the dataset.

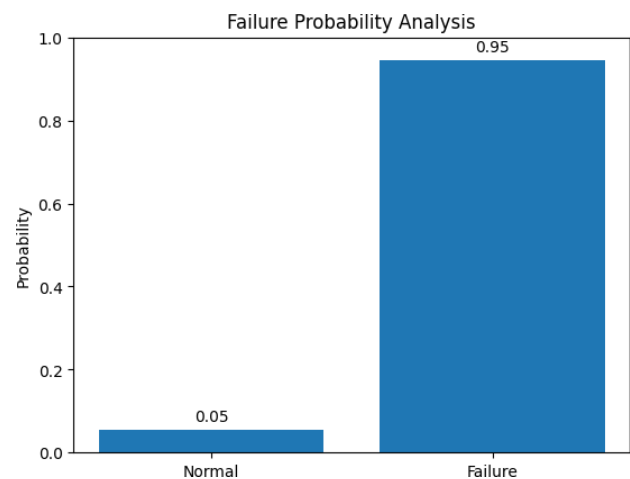
The use of SMOTE-ENN helped improve the detection of failure cases by balancing the dataset, while threshold adjustment ensured that the model maintained a suitable balance between precision and recall. The evaluation results indicate that the proposed system is reliable and suitable for real-world predictive maintenance applications.

Overall, the study shows that a well-designed machine learning pipeline can effectively identify

potential machine failures and support proactive maintenance strategies.

A detailed analysis of the confusion matrix indicates that the model is able to correctly classify a large number of normal instances while maintaining a strong detection rate for failure cases. The relatively low number of false negatives suggests that the model rarely misses actual failures, which is essential for predictive maintenance systems. Although some false positives are present, they are acceptable in practical applications as they lead to precautionary inspections rather than unexpected breakdowns.

The precision-recall curve further highlights the model's behavior under different threshold settings. As recall increases, precision decreases, reflecting the trade-off between detecting more failures and reducing false alarms. The selected threshold provides a balance that prioritizes failure detection while keeping false positives within an acceptable range.



The failure probability analysis shown in Fig. illustrates the model's prediction for a sample input. The model assigns a high probability to the failure class, indicating a strong likelihood of machine failure. This probabilistic output allows maintenance decisions to be made based on risk levels rather than only binary predictions.

Such analysis is useful in real-world applications, where maintenance actions can be prioritized depending on the severity of the predicted risk. The ability to quantify failure probability enhances the practical usefulness of the predictive maintenance system.

Feature importance analysis shows that variables related to mechanical load and wear have a significant impact on predictions. This aligns with engineering understanding, where excessive stress and prolonged usage contribute to machine degradation. These findings validate both the model and the feature construction approach used in this study.

CONCLUSION

This study presented a machine learning-based approach for predictive maintenance using the AI4I 2020 dataset. The objective was to develop a system capable of identifying machine failures in advance and supporting maintenance planning. A structured workflow was followed, including data preprocessing, feature construction, handling class imbalance, model development, and performance evaluation. One of the main challenges in the dataset was the imbalance between normal and failure instances. This issue was addressed using the SMOTE-ENN technique, which improved the model's ability to learn from failure cases. In addition, new features such as temperature difference, power, and wear rate were introduced to better represent machine behavior and improve prediction accuracy. Several machine learning models were evaluated, including Logistic Regression, Support Vector Machine, Random Forest, Extra Trees, and CatBoost. Among these, CatBoost provided the most consistent performance. After tuning and threshold adjustment, the model achieved a strong balance between recall and precision, making it suitable for predictive maintenance applications. The high ROC-AUC score further confirmed the model's ability to distinguish between normal and failure conditions. The results show that combining data balancing, feature construction, and advanced machine learning techniques can significantly improve failure prediction. The proposed system is capable of detecting potential failures at an early stage, which helps reduce unexpected downtime and supports proactive maintenance strategies. Overall, the study demonstrates that machine learning can be effectively applied to industrial datasets for predictive maintenance. The developed framework is practical, scalable, and capable of assisting decision-making in real-world environments.

Practical Impact

The proposed predictive maintenance system can be applied in real industrial environments to support maintenance planning and decision-making. By identifying potential failures in advance, the system helps reduce unplanned downtime and maintenance costs. It also improves operational safety by allowing early intervention before critical failures occur.

The approach can be integrated with existing monitoring systems and adapted to different types of machinery. This makes it a scalable and practical solution for industries adopting data-driven maintenance strategies.

Limitations:

Although the proposed system shows strong predictive performance, certain limitations should be considered. The model is trained on a specific dataset and may require adaptation when applied to different industrial environments. Variations in machine types, operating conditions, and sensor configurations can affect model performance.

In addition, the system relies on the quality of input data. In real-world scenarios, sensor data may contain noise, missing values, or inconsistencies, which can impact prediction accuracy. Addressing these issues may require additional preprocessing and data validation techniques.

Finally, while the model provides accurate predictions, further work is needed to improve interpretability so that maintenance engineers can better understand the reasoning behind each prediction.

FUTURE WORK

While the proposed system shows promising results, there are several directions in which this work can be extended. One possible improvement is to incorporate real-time data from sensors so that the system can continuously monitor machine conditions. This would allow predictions to be updated dynamically and make the system more suitable for practical industrial use.

Another area for further exploration is the use of more advanced learning techniques that can analyze sequential or time-dependent data. These approaches may capture changes in machine behavior over time

more effectively, especially when continuous monitoring data is available.

In addition, efforts can be made to improve the interpretability of the model. Providing clear explanations for predictions can help engineers better understand the reasons behind potential failures and support more informed maintenance decisions.

Future work can also focus on integrating the predictive model into a user-friendly interface or dashboard. This would make it easier for operators to visualize machine conditions, track risk levels, and take appropriate actions without requiring technical expertise.

Overall, extending the system in these directions can enhance its practical applicability and make it more valuable for real-world industrial environments.

REFERENCES

1. A. Saxena and K. Goebel, "Turbofan Engine Degradation Simulation Data Set," NASA Ames Prognostics Data Repository, 2008.
2. S. Zhang, Y. Zhang, and Q. Wang, "A review of machine learning in predictive maintenance," *IEEE Access*, vol. 7, pp. 123456–123470, 2019.
3. J. Lee, H. Davari, J. Singh, and V. Pandhare, "Industrial Artificial Intelligence for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 18, pp. 20–23, 2018.
4. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
5. L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
6. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
7. A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. O'Reilly Media, 2019.
8. 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.
9. I. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.
10. L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
11. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
12. J. Brownlee, "Imbalanced classification with Python: Better metrics, balance skewed classes," *Machine Learning Mastery*, 2020.
13. A. Kusiak and A. Verma, "A data-driven approach for monitoring blade pitch faults in wind turbines," *IEEE Trans. Sustainable Energy*, vol. 2, no. 1, pp. 87–96, 2011.
14. S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, no. 3, pp. 249–268, 2007.
15. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

HOW TO CITE: Vinayak S. Deshmukh, Pari Dargude*, Priyanshu Dhokane, Aditya Dhurve, Om Date, Mukund K. Nalawade, *Artificial Intelligence And Machine Learning- Based Predictive Maintenance Of Industrial Machines*, *Int. J. Sci. R. Tech.*, 2026, 3 (5), 601-613. <https://doi.org/10.5281/zenodo.20257164>