International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 2, Mar-Apr 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Machine Failure Prediction Using Machine Learning: A Multi-Stage Approach for Predictive Maintenance

Omkar Singh[1], Manpreet Hire[2], Om Patel[3], Sahil Singh[4]
[1]H.O.D. (Data Science), [2]Asst. Prof. (Data Science), [3 4]P.G. Student (Data Science)
[1 2 3 4]Thakur College of Science and Commerce, Mumbai, India - 400101
[1]omkarsingh@tcsc.edu.in, [2]manpreet0307@outlook.com, [3]ompatel2587@gmail.com, [4]sahilsingh2004cs@gmail.com

----------**********----------

Abstract:

The research here provides a predictive maintenance framework that uses machine learning to detect early failure in industrial equipment. A two-stage classification method is used, where the first predicts whether failure is imminent and the second predicts the exact type of failure. Feature engineering methods like power calculation and temperature difference analysis are used to improve the performance of the model. Different machine learning models are tested, and their performance is compared with accuracy, precision, recall, and F1-score. Model predictions are examined for interpretability and transparency using an explainability method, which identifies important factors affecting failure predictions. The approach boosts maintenance efficiency through the minimization of unplanned downtime and maximization of maintenance scheduling, which leads to improved industrial reliability and cost reduction.

Keywords — **Predictive Maintenance, Machine Failure, Classification**

----------**********----------

I. INTRODUCTION

Predictive maintenance (PdM) is a data-driven method that enables industries to foresee equipment failures prior to their occurrence, minimizing unplanned downtime and maintenance expenses. Contrary to reactive maintenance, where failures are responded to after their occurrence, or preventive maintenance, where maintenance occurs on a pre-set schedule, PdM utilizes real-time sensor data and analytics to refine maintenance schedules. Through the monitoring of such critical machine parameters like temperature, speed of rotation, torque, and tool wear, PdM facilitates a transition from conventional maintenance practices to a more effective, condition-based practice. Not only does this increase equipment reliability but it also minimizes downtime and maintenance expenses. Industries like manufacturing, energy, and transportation more and more rely on predictive maintenance to enhance operating efficiency and asset life. Despite this, difficulties like data quality, class imbalance, and interpretability of the model need to be overcome in order to build accurate and consistent failure prediction systems. As industrial systems become increasingly complex, proper predictive maintenance solutions are a necessity to reduce disruption and maximize resource utilization.

II. LITERATURE REVIEW

Ali Hosseinzadeh et al. investigated an AI-based predictive maintenance method for failure prediction in manufacturing by employing ML, DL, and DHL models. Comparing the performance of different models on a synthetic dataset, LightGBM outperformed all with an accuracy of 93%, whereas DL models were hindered by complicated parameter tuning. The research points to challenges such as dataset imbalance and real-world validation, highlighting the potential of AI to minimize downtime and enhance reliability but requiring additional research to increase model robustness. Although promising results are achieved, future research should aim at enhancing generalizability and tackling practical deployment issues.[1]

Ghasemkhani et al. put forward the Balanced K-Star technique to address imbalance in datasets of predictive maintenance in IoT-facilitated industries. Through modification of the K-Star algorithm through Bayesian inference, the model yielded 98.75% accuracy on the AI4I 2020 dataset, outpacing conventional ML methods. The research highlighted explainable AI (XAI) requirements in industrial contexts and specified main failure determinants such as torque and tool wear. Despite the success, challenges in computational complexity and scalability were identified, and future work being directed towards real-time implementation as well as hybrid AI models to enhance predictive maintenance.[2]

Kusumaningrum et al. utilized machine learning for predictive maintenance in smart manufacturing to minimize downtime and maximize reliability. From industrial multi-sensor data, they employed SVM and Random Forest (RF) to diagnose machine conditions and forecast Remaining Useful Life (RUL). RF performed better than SVM with 97.4% diagnostic and 89.6% prognostic accuracy. The research highlighted ML's potential in maximizing maintenance while reporting issues such as data noise and parameter tuning. Subsequent studies must develop more accurate predictive models and investigate other methods for wider industrial use.[3]

Bouabdallaoui et al. utilized machine learning for predictive maintenance in building infrastructure to avoid breakdowns and enhance efficiency. By leveraging IoT sensors and a Building Automation System, they created an LSTM-based autoencoder to identify anomalies in HVAC systems. An example in a sports facility revealed efficient failure prediction, although issues such as data availability and false alarms persisted. The research highlighted ML's potential to minimize maintenance expenditure and energy loss, recommending areas of future study in model scalability, enhanced data approaches, and wider facility utilization.[4]

Lee et al. discussed the application of AI methods, such as SVM, RNN, and CNN, for predictive maintenance (PdM) in manufacturing, and particularly machine tools. They used these models to monitor spindle motors and cutting tools and predict tool wear and bearing faults. The research indicated that CNN models were superior in identifying bearing defects and that SVM was good for tool wear monitoring. The study emphasized the potential of AI to maximize maintenance, minimize downtime, and maximize equipment life, but mentioned challenges such as data preprocessing and hyperparameter tuning. The future research needs to improve models and investigate hybrid methods for enhanced accuracy.[5]

Paolanti et al. (2018) suggested a machine learning-driven predictive maintenance (PdM) technique based on Random Forest classifiers to process sensor data from electrical motors and machinery. Deployed in an Industry 4.0 environment on Azure Machine Learning Studio, the model was able to predict failures at 95% accuracy from more than 530,000 sensor readings. The research highlighted the advantages of PdM over preventive maintenance through the ability to detect faults early. Nonetheless, difficulty in data preprocessing, feature engineering, and hyperparameter optimization were reported, and future research is aimed at enlarging datasets and optimizing models to achieve better accuracy in different industrial applications.[6]

III. METHODOLOGY

Logistic Regression:

An uncomplicated statistical model for binary classification, predicting a probability for a binary result by modeling the logistic curve. It is effective and easy to interpret but tends to be unsuitable for very complicated or non-linear relationships. It performs well with linear separable data and is simple to apply.

Random Forest:

An ensemble algorithm that constructs several decision trees and returns the most common vote. It is overfitting-resistant and great for big data, supporting categorical and numerical attributes. It also gives some information regarding feature importance. Random Forest supports missing values nicely.

Gradient Boosting:

One technique for ensemble learning where models are created sequentially in such a manner that each is learning from and improving upon the other. Successful in dealing with complicated, non-linear relationships among data, high prediction accuracy models like XGBoost Gradient Boosting methods fare well with complex data types also. It tends to be very sensitive to noise but works properly if tuned accurately.

K-Nearest Neighbors (KNN):

A learning algorithm using instances, which will classify points as a function of proximity to nearby points. It is not rigid but may be computationally costly with high dimensional or large datasets. There is no training phase for KNN but it may have difficulties scaling up with big datasets.

Multi-Layer Perceptron (MLP):

A type of neural network with multiple layers of neurons. MLP is powerful for modeling non-linear relationships and multi-label classification tasks but requires significant computational resources and careful tuning to avoid overfitting. It excels at handling complex patterns but may require large datasets to train effectively.

Support Vector Machine (SVM):

A binary classification supervised learning algorithm that identifies the hyperplane dividing the data with the largest margin. It works well for smaller datasets with widely separated classes but can be less effective with complex or noisy data. Linear SVM is especially valuable for high-dimensional data and produces stable classification outputs.

XGBoost:

A tuned gradient boosting algorithm optimized for speed and performance. It features regularization to avoid overfitting and performs exceptionally well with large complex datasets, and it is perfectly suited for binary as well as multi-label classification problems.

Dataset Description

The data employed in this research was sourced from Kaggle and is synthetically generated to mimic an industrial environment. The dataset mimics actual machine operating conditions, with sensor measurements and machine parameters that are pertinent to predictive maintenance. The dataset contains essential machine features including:

- Temperature – It measures heat fluctuations in machine parts.
- Torque – Reflects mechanical stress and load conditions.
- Rotational Speed – Monitors variations in machine speed that could indicate problems in operations.
- Tool Wear – Traces the wearing out of machine tools over a period of time.
- Power Output – Reflects efficiency and possible performance problems.

The data set also includes failure labels, specifying whether there is a machine failure and the type of failure (e.g., tool wear failure, heat dissipation failure, power failure, overstrain failure, or random failures). Table 1 Dataset Features and Units displays the key machine parameters used in the study along with their respective measurement units.

Feature	Type	Unit
Product Type	Categorical	-
Air temperature	Continuous	Kelvin (K)
Process temperature	Continuous	Kelvin (K)
Rotational speed	Discrete	RPM
Torque	Continuous	Newton meter (Nm)
Tool wear	Discrete	Minutes (min)

Table 1: Dataset Features and Units

The dataset, though synthetic, is constructed to simulate industrial conditions, thus can be used to test machine learning models for predictive maintenance. Synthetic data can introduce bias or miss some of the complexities present in real industrial systems, so rigorous preprocessing and validation procedures are performed to improve its credibility. Table 1 presents the dataset characteristics, their descriptions, and the units of measurement for each parameter.

Workflow of the System

The proposed methodology consists of two key stages: machine failure prediction and failure type identification. In Stage 1, a binary classification model is implemented to predict whether a machine will fail or continue operating normally. Various machine learning algorithms are explored to develop this model, and its performance is assessed using various evaluation metrics. If a failure is detected, the process moves to Stage 2, where a multi-label classification model is used to identify the specific failure type(s). The failure types considered include Tool Wear Failure (TWF), Heat Dissipation Failure (HDF), Power Failure (PWF), Overstrain Failure (OSF), and Random Failure (RNF). Since a machine can exhibit multiple failure types simultaneously, multi-label classification techniques are applied. Figure 1: Workflow of Predictive Maintenance System illustrates the step-by-step process of failure prediction and classification.

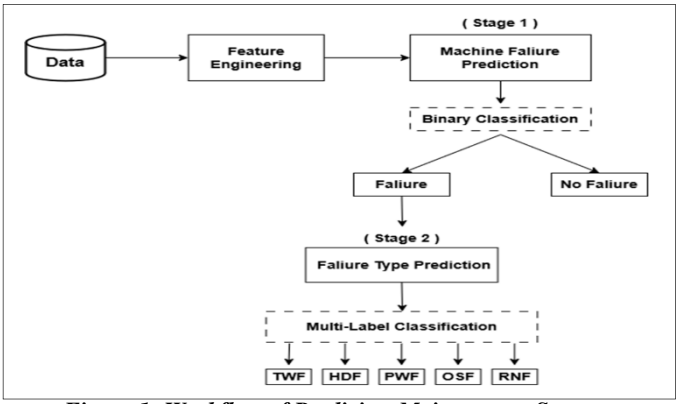

Figure 1: Workflow of Predicitve Maintenance System

Data Preprocessing

To ensure the dataset is clean and optimized for machine learning models, several preprocessing steps are applied. Since the dataset is synthetically generated, missing values are minimal. However, any gaps in sensor readings are handled using mean or median imputation for numerical features to maintain data consistency. Duplicate records, if present, are identified and removed to prevent redundancy and potential bias in model training. Additionally, sensor data may contain anomalies due to faulty readings or extreme operating conditions. To address this, outliers are detected and treated using the Interquartile Range (IQR) method, which filters extreme values, and box plots are used for visual inspection to identify data points that significantly deviate from normal operational conditions. These preprocessing steps enhance data quality, ensuring more reliable and accurate predictive modelling.

Feature Engineering

Prior to model training, extensive feature engineering was applied to enhance the predictive performance of the models. The dataset initially included raw machine parameters such as air temperature, process temperature, rotational speed, torque, and tool wear. To extract more meaningful insights from these parameters, several additional features were engineered:

- Power Calculation: Computed as the product of Torque × Rotational Speed (rad/s), capturing the mechanical load and energy consumption of the machine.
- Temperature Difference: Calculated as the difference between process temperature and air temperature, helping monitor the machine's cooling efficiency and heat dissipation.
- Rotational Speed Conversion: Rotational speed was converted from RPM (revolutions per minute) to rad/s (radians per second) to ensure consistency in units and improve model performance.

Categorical features, such as machine type, were encoded using Ordinal Encoding, ensuring that the relationships between the categories were preserved during model training. Numerical features were then standardized using StandardScaler, which normalizes the data by transforming it to have a mean of zero and a standard deviation of one. This step improves the performance of many machine learning models, especially those sensitive to feature scaling.

To address class imbalance, which is common in predictive maintenance datasets (where failure events are less frequent than normal operations), stratified sampling was applied to ensure balanced representation of both classes in the training and test sets. Additionally, class weight adjustments were made during model training to prevent bias toward the majority class, ensuring that the model is equally sensitive to predicting both failure and non-failure events.

Model Selection

	Stage 1	Stage 2
Stages	Binary classfication	Multi-Label classifcation
Model 1	Logistic Regression	Random Forest
Model 2	Random Forest	Gradient Boosting
Model 3	KNN	Multi-Layer Perceptron
Model 4	SVM	XGBoost

Table 2: Two-Stage Classification Model Selection

These models were chosen based on their ability to handle the specific tasks efficiently and their expected performance in terms of accuracy and scalability for the predictive maintenance application. Table 2: Two-Stage Classification Model Selection describes the models chosen for each stage, outlining the binary classification models used in Stage 1 and the corresponding multi-label classification models in Stage 2.

Model training begins after data preprocessing, where the dataset is split into training and testing sets. During training, the model adjusts its parameters to minimize prediction errors. The testing set is kept separate and is used to evaluate the model's performance on unseen data, ensuring that it generalizes well to real-world scenarios.

Model Evaluation

Stage 1: Predicting Machine Failure

The first step of the predictive maintenance process employs binary classification to predict a machine failure to facilitate proactive maintenance to minimize downtime as well as operational risks. The model performance is evaluated based on several critical metrics. Accuracy provides overall accuracy but can be misleading in the presence of class imbalance. Precision ensures that predicted failures are actually failures, minimizing false alarms and unnecessary maintenance. Recall evaluates the proportion of actual failures correctly predicted, a critical aspect in early warning and surprise breakdown prevention. F1-Score provides a

balance between precision and recall, a benefit in the presence of imbalanced datasets. ROC-AUC score evaluates the model's ability to distinguish between failure and non-failure instances, with larger values indicating better classification performance. The confusion matrix provides insights into misclassification patterns by breaking down true positives, true negatives, false positives, and false negatives.

Stage 2: Failure Type Prediction (Multi-Label Classification)

During the second stage, micro-averaged F1-score was utilized as the primary evaluation metric to achieve an unbiased and fair measure of how well the model is performing to predict various types of failures while accounting for class imbalance. Micro-Averaged F1-Score Averages true positives, false positives, and false negatives across all the types of failures before computing the F1-score. It is a superior option for imbalanced data as it yields a class distribution-weighted measure of performance.

Additionally, training and inference times were compared across different models to assess computational efficiency. A model's ability to provide quick predictions is essential for real-time maintenance applications, and these time metrics help in identifying the best-performing model in terms of speed and accuracy.

Model Interpretation

LIME (Local Interpretable Model-agnostic Explanations) is an interpretability method that provides explanations for individual machine learning model predictions. It perturbs input and examines the manner in which changes in model outputs reflect changes in input in order to determine the most salient features. LIME is model-agnostic, i.e., it can be used on any black-box model, thus making it appropriate for explaining intricate decision-making. It brings about transparency, trust, and debugging through locally faithful explanations, and it is thus feasible to use it on predictive maintenance in order to appropriately understand failure prediction.

To increase the explainability of models, the LIME tool (Local Interpretable Model-agnostic Explanations) was used for explaining the feature contributions to predictive maintenance. LIME disturbs input data and observes changes in predictions, providing locally faithful explanations.

At Stage 1 (Binary Classification), LIME identified key features in failure prediction, for instance, temperature difference, power difference, and tool wear. At Stage 2 (Multi-label Classification), it also revealed key parameters for different failure modes, for instance, TWF, PWF, and OSF.

Through the use of LIME, the model's decision-making process was more understandable, enhancing trust, debugging, and actionable knowledge. Figure 2 illustrates the LIME explanation of the sample input used.

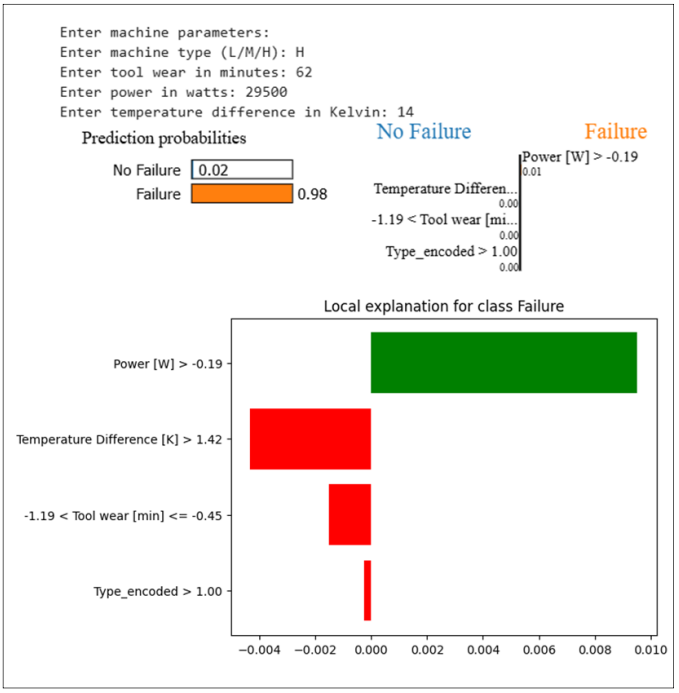

Figure 2: LIME Explanations for Feature Importance in Predictive Maintenance

Results

AUC ROC Curve and Score

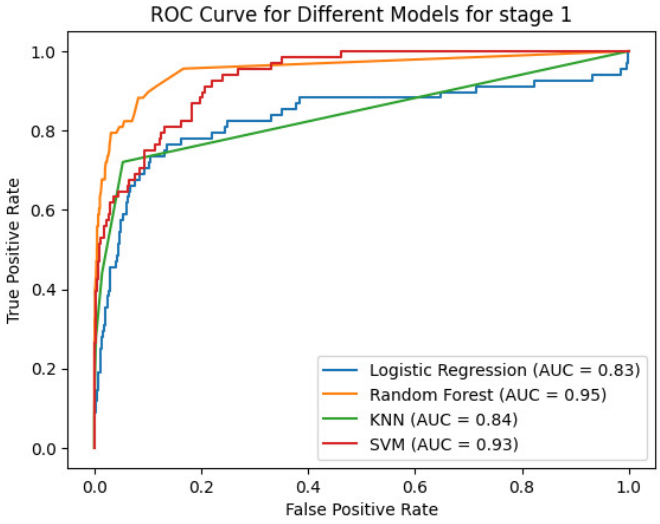

Figure 3: ROC-AUC Curve and Scores for Stage 1 Models

In Phase 1, the AUC-ROC curve and AUC score were employed to measure model performance in separating non-failure and machine failure. A larger AUC (nearer to 1) reflects greater classification capability. Random Forest had the best AUC score of 0.95, followed by SVM (0.93), while KNN (0.84) and Logistic Regression (0.83) performed lower. These outcomes indicate that more complex models such as Random Forest and SVM perform better in identifying failure patterns compared to more basic models. Figure 2 plots the comparative AUC-ROC curves of each model.

Micro F1 Score

Stage 2	Multi-Label Classification	Micro F1 Score
Model 1	Random Forest	0.967
Model 2	Gradient Boosting	0.978
Model 3	Multi-Layer Perceptron	0.973
Model 4	XGBoost	0.974

Table 3: Micro F1 Score Comparison for Stage 2 Models

In Phase 1, the AUC-ROC curve and AUC score were employed to measure model performance in separating non-failure and machine failure. A larger AUC (nearer to 1) reflects greater classification capability. Random Forest had the best AUC score of 0.95, followed by SVM (0.93), while KNN (0.84) and Logistic Regression (0.83) performed lower. These outcomes indicate that more complex models such as Random Forest and SVM perform better in identifying failure patterns compared to more basic models. Figure 2 plots the comparative AUC-ROC curves of each model.

Training and Inference Time Overview

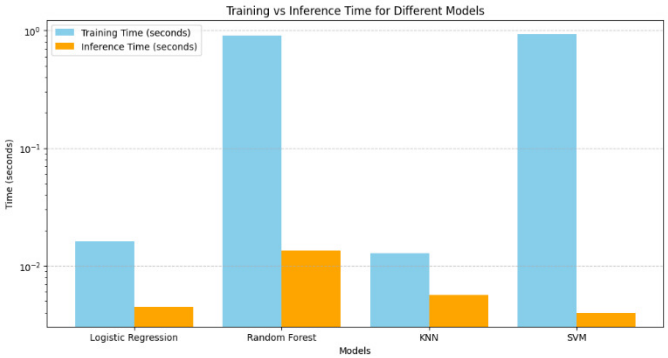

Figure 4: Training and Inference Time Comparison for Binary Classification Models

During Stage 1, both accuracy and computational efficiency, as well as training and inference times, were considered in model evaluation. Logistic Regression trained and inferred the fastest but with lower predictive accuracy. Random Forest and SVM took longer to train because of their complexity but had greater accuracy. KNN trained rapidly, just like Logistic Regression, but with lower predictive accuracy compared to Random Forest and SVM. These findings demonstrate the trade-off between model performance and computational efficiency. Figure 4 is the Training and Inference Time Comparison for Binary Classification Models.

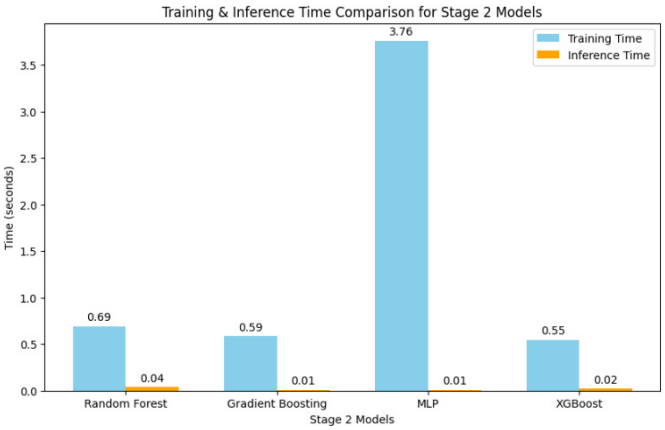

Figure 5: Training and Inference Time Comparison for Multi-Label Classification Models

In Stage 2, multi-label classification models were compared based on training and inference times. XGBoost took the least time to train and was thus most efficient at learning, while Gradient Boosting struck a balance between speedy training and low inference time, making it most suitable for real-time use. MLP took the longest time to train but produced fast inference, guaranteeing efficient prediction upon training. Random Forest was moderately time-consuming during training but took the longest during inference, and hence it is not ideal for quick decision-making. Figure 4 shows the Training and Inference Time Comparison for Multi-Label Classification Models.

IV. CONCLUSIONS

The two-stage predictive maintenance architecture places emphasis on selecting models as per operational requirements.

- Logistic Regression → Random Forest: Ideal for quick model updates in environments with limited resources, great for quick prototyping.
- Random Forest → Gradient Boosting: Applicable to high-accuracy critical applications (aerospace, medicine), where missed positives are expensive.

- SVM → XGBoost: Ideal for high-volume manufacturing, balancing between precision and computation time.
- KNN → MLP: Best for in-depth failure analysis, ideal for environments where there are complex failure modes.

The sequential nature of the framework emphasizes the importance of Stage 1 accuracy, as any misclassifications will affect Stage 2 results. Therefore, selection should prioritize strong Stage 1 performance while considering the specific requirements and constraints of the maintenance environment.

V. FUTURE WORK

This research brings out the potential of a two-stage predictive maintenance model, yet there are aspects that can be worked upon further. Future research can be oriented towards incorporating actual industrial data for model performance verification in real-world scenarios, working on issues like sensor noise, missing values, and domain-dependent failure patterns. More sophisticated feature engineering, with domain-dependent features like vibration patterns, energy consumption patterns, and contextual machine usage information, would enhance predictive precision. Improving model interpretability with tools such as SHAP, LIME, or counterfactual explanations can enhance trust in AI-driven decisions so that they are more actionable for engineers. In addition, investigating adaptive and self-learning models using online learning and reinforcement learning can enable models to dynamically adapt to changing machine conditions and failure patterns. Improving these areas will make predictive maintenance systems more accurate, scalable, and adaptive, ultimately enhancing industrial efficiency and minimizing downtime.

REFERENCES

[1] A. Hosseinzadeh, F. F. Chen, M. Shahin, and H. Bouzary., "An AI-Based Predictive Maintenance Method for Failure Prediction in Manufacturing Using Machine Learning, Deep Learning, and Deep Hierarchical Learning Models," Journal of Manufacturing Processes, vol. 80, pp. 1–12, 2023.

[2] B. Ghasemkhani, Ö. Varlıklar, and D. Birant, "Balanced K-Star: An Explainable Machine Learning Method for Internet-of-Things-Enabled Predictive Maintenance in Manufacturing," Machines, vol. 11, no. 3, p. 322, 2023, doi: 10.3390/machines11030322.

[3] D. Kusumaningrum et al., "Machine Learning for Predictive Maintenance in Smart Manufacturing: Minimizing Downtime and Maximizing Reliability," Procedia CIRP, vol. 105, pp. 123–128, 2022.

[4] Y. Bouabdallaoui et al., "Machine Learning for Predictive Maintenance in Building Infrastructure: Avoiding Breakdowns and Enhancing Efficiency," Journal of Building Performance, vol. 13, no. 3, pp. 345–360, 2022.

[5] J. Lee et al., "Application of AI Methods for Predictive Maintenance in Manufacturing: A Case Study on Machine Tools," Journal of Manufacturing Science and Engineering, vol. 142, no. 6, p. 061009, 2020.

[6] M. Paolanti et al., "Machine Learning-Driven Predictive Maintenance Technique Based on Random Forest Classifiers for Industrial Applications," in Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2018, pp. 1021–1025.

[7] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the Industry 4.0: A systematic literature review," Computers & Industrial Engineering, vol. 150, p. 106889, 2020, doi: 10.1016/j.cie.2020.106889.

[8] G. M. Sang, L. Xu, P. de Vrieze, Y. Bai, and F. Pan, "Predictive Maintenance in Industry 4.0," in Proceedings of ICIST 2020, Lecce, Italy, June 2020, doi: 10.1145/3447568.3448537.

[9] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. G. Jun, and J. W. Sutherland, "Predictive Maintenance of Machine Tool Systems Using Artificial Intelligence Techniques Applied to Machine Condition Data," Procedia CIRP, vol. 80, pp. 506–511, 2019, doi: 10.1016/j.procir.2018.12.019.

[10] P. Nunes, J. Santos, and E. Rocha, "Challenges in predictive maintenance – A review," CIRP Journal of Manufacturing Science and Technology, vol. 40, pp. 53–67, 2023, doi: 10.1016/j.cirpj.2022.11.004.

