

Predictive Maintenance Testing in Machine Learning: Combining Manual Insights, Java Programming and Data Science for Automation

Daniel Raj Jeevaguntala*

Department of Health, Division of Operation and Systems, State of New York, USA

Citation: Jeevaguntala DR. Predictive Maintenance Testing in Machine Learning: Combining Manual Insights, Java Programming and Data Science for Automation. *J Artif Intell Mach Learn & Data Sci* 2025, 3(1), 2417-2426. DOI: doi.org/10.51219/JAIMLD/daniel-raj-jeevaguntala/521

Received: 24 February, 2025; **Accepted:** 05 March, 2025; **Published:** 07 March, 2025

***Corresponding author:** Daniel Raj Jeevaguntala, Department of Health, Division of Operation and Systems, state of New York, USA, E-mail: Danielrajjeevaguntala1@outlook.com

Copyright: © 2025 Jeevaguntala DR., This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABSTRACT

A predictive maintenance approach has developed as a revolutionary means to minimize downtime and operational costs in industrial systems. Machine learning techniques are used in this study to design a robust predictive maintenance framework for potential equipment failures based on relevant operational parameters. Random Forest and XGBoost models were trained and evaluated using the AI4I 2020 dataset and the machine failure prediction is highly accurate. Based on domain expertise, these key features, namely Torque [Nm] and Tool Wear [min], were identified as pivotal indicators since they can provide actionable insights for maintenance teams within the domain. The analysis of the two models shows XGBoost performing better on imbalanced data and predicting minority failure cases.

In addition to providing predictive accuracy, the study offers practical deployment strategies, including saving models with joblib, implementing Java GUIs for real-time user interaction and automating workflows using APIs and task schedulers. Moreover, this research bridges the gap between theoretical machine learning models and practical applications and provides a scalable, user-friendly framework for industrial predictive maintenance. These insights and methodologies can optimise resource allocation, enhance decision-making and transition the industry from reactive to proactive maintenance practices. Future work involves integrating real-time data streams from IoT and advanced neural networks to increase system scalability and precision.

Keywords: Predictive Maintenance, Machine Learning, Random Forest, XGBoost, Industrial Automation, Torque Analysis, Tool Wear, Real-Time Predictions, Failure Detection, IoT Integration

1. Introduction

Predictive maintenance has been an essential way for modern-day industries to move towards more proactive and efficient systems than traditional reactive or preventive maintenance methods¹. This is in contrast to reactive maintenance, which addresses equipment failure after occurrence and preventive maintenance, which is based on scheduled servicing and relies on data-driven insights for predictive maintenance to predict failures well ahead of occurrence. Such a paradigm shift

dramatically cuts down on unplanned downtime, improves operational efficiency and reduces maintenance costs².

Continuous and reliable equipment performance is significant for manufacturing, aircraft and energy production industries. The incidence of such unplanned disruption is not only a financial loss but also a breach of safety standards, which can be compromised. This is essentially where predictive maintenance steps in to mitigate these risks by using real-time sensor data from machine learning-based models to predict the

machine's potential failure³. With this approach, maintenance teams can proactively take care of the issues, thereby extending the lifespan of equipment. Moreover, the increased adoption of Industry 4.0 principles also makes integrating predictive maintenance into an automated system necessary for attaining operational excellence⁴.

Although the benefits of predictive maintenance are acknowledged everywhere, existing maintenance strategies are still lacking when facing real-world challenges. While reactive maintenance is straightforward, it often incurs costly downtimes and safety risks. Although systematic, preventive maintenance is inefficient as it uses fixed schedules that may not fit with the actual equipment conditions. Industrial applications need this flexibility or adaptability, but neither strategy can provide it⁵.

Additionally, the integration of predictive maintenance into real-time systems has been more limited. Machine learning tools have strong power-based tools to analyze sensor data and forecast failures, but the misalignment with domain expertise and manual insights hinders their practical use⁶. In addition, most of the studies on predictive maintenance have been done using isolated models or datasets and comprehensive comparative analyses on machine learning algorithms engineered to suit industrial needs have not been done⁷. The lack of research for algorithms such as Random Forest and XGBoost, two of the most effective algorithms we know for predictive tasks, which are underexplored in predictive maintenance, speaks to the need for rigorous evaluations of such algorithms. This gap must be filled to enable systems that create actionable and realistic maintenance insights integrating machine learning, manual expertise and automation.

This study aims to construct a robust predictive maintenance framework for improved automation using machine learning algorithms, manual insights and Java programming. This framework attempts to overcome the shortcomings of current maintenance strategies in predicting equipment failures accurately and in real time by analyzing data from industrial systems.

For this purpose, this study will evaluate the performance of two widely used machine learning algorithms, Random Forest and XGBoost, on the AI4I 2020 Predictive Maintenance dataset. Preprocessing the data set, training and evaluating the sample and finding features most important for accurate failure prediction are subject to the scope. A pipeline for integrating these models into an automated pipeline, providing seamless real-time predictions, is also studied. A practical and scalable solution to predictive maintenance in industrial environments is being pursued by combining data science techniques with domain-specific knowledge and automation tools.

This research develops multiple essential contributions to the field of predictive maintenance. Firstly, it compares the Random Forest and XGBoost models for the first time on the AI4I 2020 Predictive Maintenance dataset, providing insights into the performance and applicability. In addition to assessing these models for accuracy and efficiency, the study identifies key features like torque, tool wear and process temperature, which drive equipment failures.

The research shows how machine learning models' feature-important insights improve maintenance decisions. The study then integrates these insights in the context of Java-based

automation tools. It presents a framework that is both usable and scalable for use in real-world applications. Finally, this research closes the gap by presenting a holistic method combining machine learning, human insights and automation to build more efficient and practical predictive maintenance systems.

2. Literature Review

2.1. Predictive maintenance approaches

Industries have relied on traditional maintenance strategies to retain equipment functionality, yet these strategies are inherently limited, which has led to the need for predictive maintenance systems⁸. Preventive maintenance is applied on a time-scheduled or fixed usage basis and the equipment is serviced on a scheduled basis, irrespective of its actual condition. However, although such an approach prevents some failures from happening, it typically causes unnecessary maintenance activities or misses some failure events, increasing operational costs and risks of downtime⁹. Conversely, condition-based maintenance is the process of monitoring specific parameters of equipment in real-time that can be used to identify its health condition. Despite improved maintenance precision over preventive approaches, it still depends heavily on hardwired threshold values and lacks advanced predictive abilities.

Predictive maintenance is a transformative shift that enables us to leverage data-driven methodologies to predict if a failure is likely to occur and before it happens. Unlike traditional strategies, root cause forecasting of equipment breakdowns uses sensor data, historical trends and environmental variables rather than probabilistic techniques¹⁰. However, integrating the machine learning algorithms allows predictive maintenance systems to learn more accurately about complex relationships between operational parameters and failure modes. For example, readings from sensors for parameters like temperature, torque and rotational speed can be analyzed to predict wear and tear, thereby reducing the possibility of unplanned downtime. This approach both cuts cost and increases safety and reliability across all industries¹¹.

2.2. Machine learning models for predictive maintenance

Machine learning, a robust failure prediction and anomaly detection technology power predictive maintenance. Random Forest and XGBoost are among many machine-learning models that have received considerable attention because of their high accuracy, interpretability and scalable performance. Decision trees-based ensemble learning technique Random Forest performs well with noisy datasets and could capture the non-linear relationship between variables. Its ability to provide feature importance scores in predictive maintenance is beneficial because domain experts can locate the key factors that influence failures. For example, the random forest algorithm has been shown to predict failure modes in manufacturing equipment using sensors that report torque and tool wear. Random Forest provides strong prediction results but requires a lot of computational resources for a large dataset¹².

As a gradient-boosting framework, XGBoost has performed well in classification and regression tasks. Initially, it generates the decision trees by iteratively optimising them for decision accuracy and avoiding overfitting. XGBoost has been used to predict failures in industrial systems using imbalanced datasets and predictive maintenance, gaining better precision on minority

classes over traditional models¹³. Even though XGBoost is quite strong, it is limited by its need to fine-tune its hyperparameters and increased computational complexity. While such models have worked well, recent studies emphasise one model in isolation, ignoring the relative advantages and disadvantages of the models as a group. On top of this, feature importance analysis is not emphasised either, impeding the interpretation of results and ultimately hindering maintenance teams from translating model predictions into actionable insights¹⁴. Furthermore, models in many studies are trained on synthetic or controlled datasets, impairs their generalizability to real-world industrial settings.

2.3. Role of automation and integration

The full potential of predictive maintenance systems depends on automation. Industry can automate failure predictions and maintenance scheduling by integrating machine learning models into real-time monitoring tools, allowing for less manual intervention and faster response times¹⁵. Several other studies have been done on building other predictive systems using Java or some other platform and they have shown their ability to make a scalable and efficient system. An example is using Java-based applications to capture sensor data, process it and then trigger a maintenance alert based on pre-defined conditions^{16,17}.

However, machine learning models are yet to be adequately integrated into such systems. Some Python-based tools, such as Flask or Fast API, ease the deployment of the model, but there is no research combining these with Java-based GUI to form complete systems. Integrating such would also allow maintenance teams to see predictions, interact with the system and supply new data for accurate time analysis¹⁸. In addition, there is a gap in the studies regarding user-friendly GUIs. Systems needed by industrial users must provide accurate predictions and a simple presentation of them. This gap can be bridged by linking Java-based GUIs with their machine-learning models to deliver easy-to-understand visualizations and actionable insights¹⁹. This integration challenge can be addressed through research and the resulting predictive maintenance systems will be more accessible and practical to industrial applications.

2.4. Research gap

Despite the significant advancements in predictive maintenance research, there remain substantial gaps in machine learning-based solutions, which presently limits this practical adoption. Most existing studies tend to predict a model or data sample in isolation without thoroughly comparing them. For example, Random Forest and XGBoost are among the most prominent algorithms in the domain²⁰. Yet, there is scarce research on a direct head comparison of the two on datasets that naturally vary across industrial scenarios.

The other gap is machine learning models with automation workflows. Predictive maintenance systems should be capable of predicting failures and transparent enough to facilitate seamless automation for real-time decision-making. Despite this, current research rarely considers how these models can be realised in automated environments, leaving a critical gap between developing end-to-end predictive maintenance pipelines²¹.

In addition, the usability of predictive maintenance systems is impeded by the absence of user-friendly interfaces to interact with them. The technical expertise of industrial practitioners to interpret complex model outputs is often limited. Therefore,

research is needed to combine machine learning models with simple GUIs or APIs so that users can easily interact with predictions and insights²². To fill this gap, highlights the necessity of designing pipelines that combine model deployment with real-time predictions and user-friendly interfaces.

3. Methodology

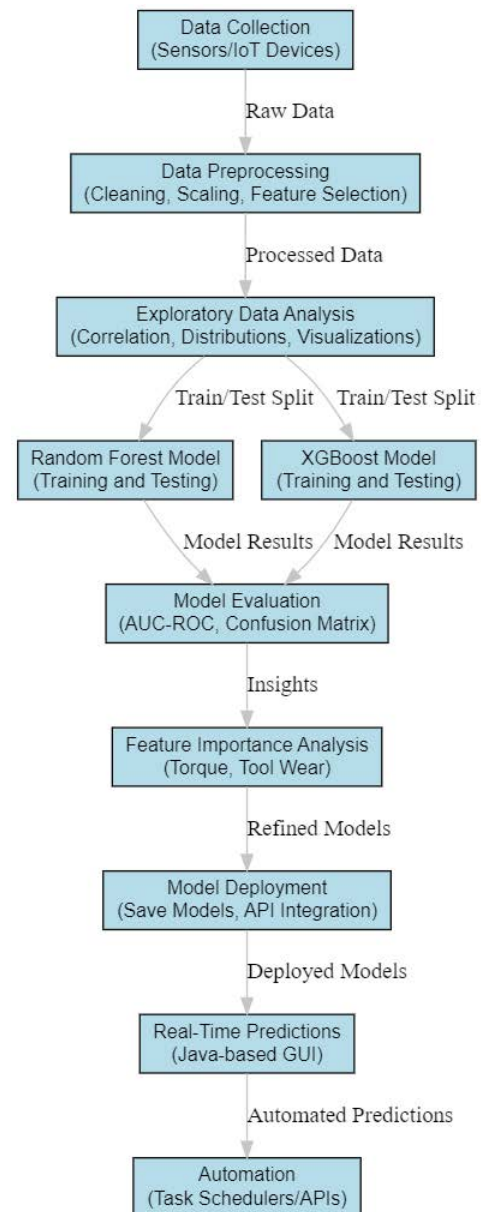


Figure 1: Proposed Methodology Diagram.

3.1. Dataset

This study uses the AI4I 2020 Predictive Maintenance dataset, a synthetic data previously constructed to simulate industrial conditions for a milling process. This includes extensive features describing operational settings, sensor measurements and machine failure data. The dataset contains 14 features and 10,000 observations, allowing its use as a balanced and diverse dataset for the development and testing of predictive maintenance models.

Air temperature [K], Process temperature [K], Rotational speed [rpm], Torque [Nm] and Tool wear [min] were key provided features. These operational quantities are essential for industrial equipment's health and performance assessment. The dataset also contains a machine failure binary variable, the target

label, for classification purposes. In addition, the five specific failure modes, Tool Wear Failure (TWF), Heat Dissipation Failure (HDF), Power Failure (PWF), Overstrain Failure (OSF) and Random Failures (RNF) are included to provide granularity in the failure analysis²³.

This dataset, relevant for predictive maintenance, realistically simulates industrial environments. This data captures the interplay between operational parameters and failure occurrences, making it a robust dataset that can be used to train machine learning models. Moreover, the presence of binary failure labels and this distinction allows for multi-level analysis and the construction of highly versatile predictive systems applied to other industries²⁴.

3.2. Data preprocessing and exploratory data analysis

Data preprocessing is a crucial step to ensure the quality and robustness of a machine learning model. The following steps were undertaken to prepare the dataset:

3.2.1. Cleaning: The missing values and duplicates were removed from the dataset. As these columns do not contribute to our predictive task, we removed non-numerical columns (UDI, Product ID, Type).

3.2.2. Feature selection: Analysis was performed on retaining key features such as temperature, torque and tool wear while removing the irrelevant attributes. This meant that the model only focused on parameters affecting machine failures.

3.2.3. Scaling: To bridge the feature size gap (e.g., rotational speed in RPM and torque in Nm), the data was standardized by adding Standard Scaler. This was an essential first step to help improve the performance of machine learning algorithms, particularly those sensitive to feature scaling.

3.2.4. Exploratory insights: Exploratory data analysis (EDA) was conducted to obtain initial insights into the dataset. It was found that a process temperature and an air temperature showed a strong correlation, which indicated how both influence machine performance. A clear separation between failure and non-failure cases on tool wear and torque was shown in pair plots. Torque and rotational speed outliers were flagged in box plots, which may indicate precursors to potential failure. These insights formed the direction of the feature importance analysis in subsequent modelling stages.

3.3. Model selection and implementation

This study used two machine learning models with proven efficacy in classification tasks and the capability to handle imbalanced datasets: Random Forest and XGBoost.

3.3.1. Random Forest: Due to robustness in handling noisy and high-dimensional data, an ensemble learning method was picked for random forest. Calculating the importance of features was helpful, as it allowed critical insights into operational parameters that affect machine failure¹. The interpretability and efficiency of operating on disparate features make the model well-suited for predictive maintenance.

3.3.2. XGBoost: The advanced boosting mechanism in XGBoost of building decision trees to optimise predictive accuracy was selected. It suited this study perfectly because it could deal with imbalanced datasets and regularise the model, thus reducing overfitting. XGBoost excelled in predicting minority failure classes in a dataset where the failure distribution is skewed⁷.

3.3.3. Implementation details: The data was split into training and testing sets to evaluate model performance, with 80% training and 20% testing. First, both models were trained with their default hyperparameters and then were fine-tuned to improve results. For Random Forest, `n_estimators` was set to 100 and maximum tree depth was adjusted for best accuracy. One hundred estimators with a learning rate of 0.1 and a max tree depth of 5 were passed to XGBoost. They processed the data and trained the models on the test set with the standard accuracy metrics.

3.4. Evaluation metrics

The performance of the models was assessed using a combination of classification metrics and visualisations:

3.4.1. Classification report: Precision, Recall and F1 were calculated to determine how well the models could correctly classify failure and non-failure cases. A Recall measure within the measurement was the proportion of actual failures the model detected and a Precision measure calculated the ratio of correctly predicted faults overall faults predicted. F1-score provided an overall measure of accuracy, harmonic mean of Precision and Recall.

3.4.2. Confusion matrix: Classification results of the models were visualised using confusion matrices. They described every number of true negatives, true positives, false positives and false negatives and explained the model performance in detail.

3.4.3. ROC-AUC: By plotting Receiver Operating Characteristic (ROC) curves, we compared the accurate favorable rates of the model vs the false positive rate across different thresholds. A higher value of the AUC of the ROC curve was taken as a comprehensive measure of the model performance. XGBoost was slightly better than Random Forest in terms of AUC and minority class prediction. These results verified model selection and set the groundwork for integrating the selected models into real-time predictive maintenance systems.

4. Results and Analysis

4.1. Exploratory data analysis

Figure 2 correlation heatmap shows how features in the dataset relate to one another. An air temperature [K] and process temperature [K] correlation of 0.88 surfaces show that these variables tend to increase together, as expected in real-world operational dynamics. Also, Torque [Nm] has a strong negative correlation (-0.88) with rotational speed [rpm], which is the case for the operation of the machine. The Tool Wear [min] vs Machine Failure correlation (0.11) would significantly predict tool wear as a potential predictor. This finding reinforces applying vital sensor data for predictive maintenance and their relation between operating variables and an indication of failures.

The feature distributions demonstrate the spread and central tendencies of definitive variables. Air Temperature [K] and Process Temperature [K] parameters have normal-like distributions, while Rotational Speed [rpm] and Torque [Nm] are more skewed, indicating diverse operational states. Tool Wear [min] 's distribution is uniform, suggesting it increases over time until failure. Thus, these insights validate the dataset as appropriate for predictive maintenance tasks. It encompasses different operating conditions that coincide with the research goal to identify failure patterns under different states (**Figure 3**).

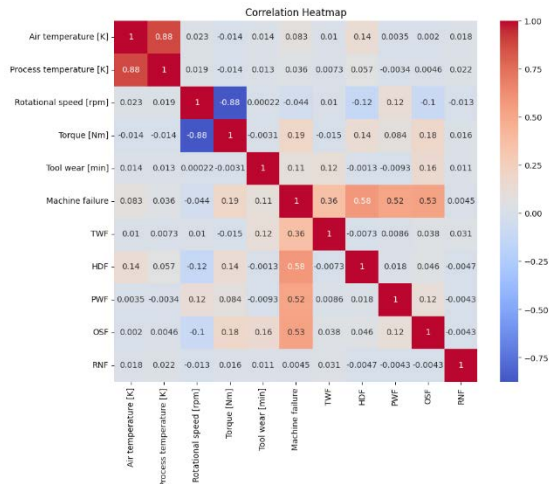


Figure 2: Correlation Heatmap.

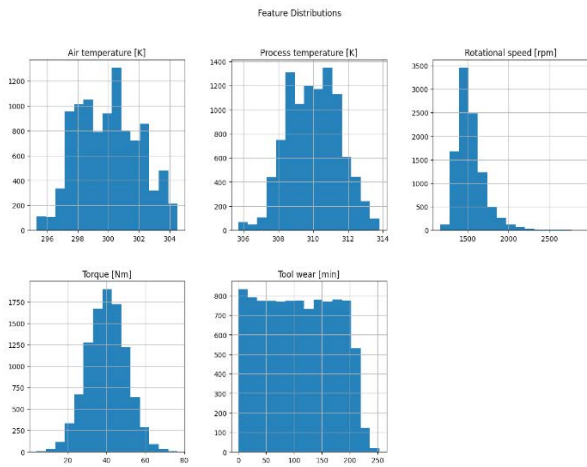


Figure 3: Feature Distribution.

The box plot indicates possible outliers in the Rotational Speed [rpm] and Torque [Nm] values, respectively, which could indicate cases that lead to machine state failures. Benchmarks for normal operating conditions are given in the median values of each parameter. For example, Tool Wear [min] has a stable interquartile range matching the fact that it provides a gradual wear indicator. The plots presented here support the objective of identifying anomalies as they can provide the early warning signal indicating potential failure that the system may proactively address through predicated maintenance interventions (Figure 4).

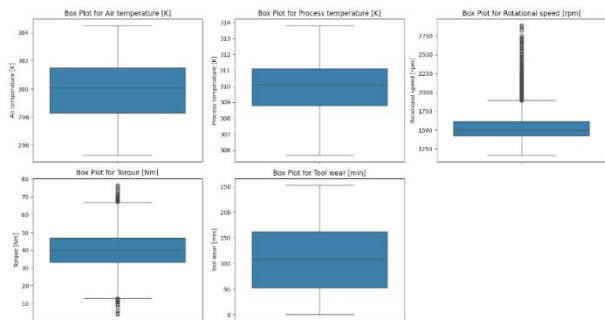


Figure 4: Box Plots.

The Machine Failure target variable is plotted against numerical features and box plots are provided to illustrate how features are separated. The clustering characteristics of failure (1) and non-failure (0) cases of Torque [Nm] and Rotational speed [rpm] indicate the importance of predicting failure. Air

Temperature [K]. Process Temperature [K]. The overlapped pattern tends to give a weaker correlation with failures, yet consistent over operations, making their ability to monitor the useful. The model shown here paints the picture set to integrate machine learning models to model these complex relationships and make failure predictions (Figure 5).

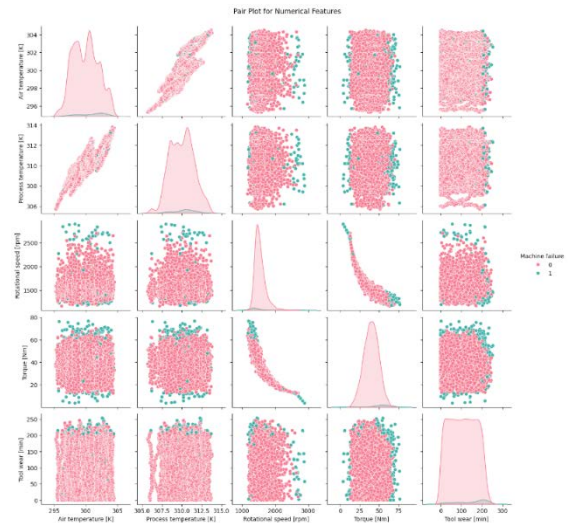


Figure 5: Pairplots.

4.2. Random forest results

The model’s trained classification report is explained and the model’s performance is based on precision, recall and F1 score (Figure 6). The model’s accuracy for class 1 (failure) is 0.82, i.e., 82% of predicted failures are accurate. However, the recall was 0.59, meaning the model did not identify 41%. The F1 score describes a precision and recall balance of 0.69. This strong performance of the model yielded an overall accuracy of 98% on correctly classifying the majority class (0), i.e., non-failure cases, which shows the accuracy of classifying the majority class. A macro average F1-score of 0.84 highlights a narrow preponderance in class handling for imbalanced classes but indicates reasonable overall predictive power in aggregate. These results follow the objective of creating a reliable predictive maintenance system; however, they demonstrate the need for additional optimization to improve recall and enable less equipment downtime.

	1	0.82	0.59	0.69	61
accuracy				0.98	2000
macro avg	0.98	0.79	0.84		2000
weighted avg	0.98	0.98	0.98		2000

Figure 6: Classification Report of Random Forest.

The detailed classification result is shown in the confusion matrix. The model correctly classified 36 failure cases (true positives) and 1,931 non-failure cases (true negatives) out of 2,000 test samples (Figure 7). Yet, it also misclassified 25 failures as non-failures (false negatives) that, if unattended, could lead to unplanned downtimes. The model was exact, with only eight non-failure cases being incorrectly predicted as failures (false positives). The model effectively identifies failure but also has limitations in identifying minority failure cases, as confirmed by the confusion matrix. The study shows that feature importance and more sophisticated algorithms, such as XGBoost, remain relevant to address such imbalances in this analysis towards the proactive failure prediction research objective.

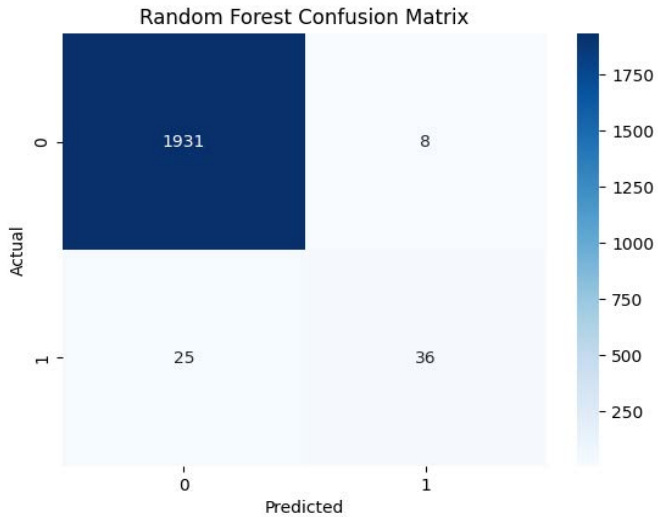


Figure 7: Random Forest Confusion Matrix.

The ROC curve is a way to depict the tradeoff between a true positive rate (sensitivity) and a false positive rate at different thresholds. The proximity to the upper left corner and the steep rise in initial suggests the model’s excellent discrimination ability to classes. Further quantification of this ability is obtained from the area under the ROC curve (AUC), with values near 1 indicating excellent classification performance (Figure 8). The model has a high AUC, which supports the objective of integrating machine learning into predictive maintenance systems to determine the validity of the model’s ability to generate reliable and actionable predictions. It also shows the model’s suitability for real-time operation in automated maintenance workflows, filling the gap in practical and scalable predictive systems.

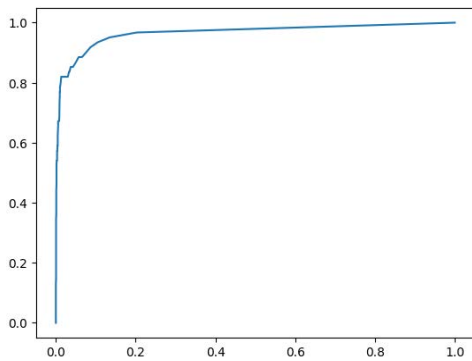


Figure 8: ROC Curve Random Forest.

As shown in the feature importance chart of Random Forest, Torque [Nm], Rotational Speed [rpm] and Tool Wear [min] were found to be the most significant parameters. These conclusions confirm the relevance of the dataset as operational indicators since torque and speed represent the mechanical stress indicators and tool wear is the indicator directly affecting equipment reliability (Figure 9). This corresponds with shifting from human inspections to using sensor data to achieve accurate failure prediction. Such critical features need to be prioritized by Random Forest as they generate actionable insights for maintenance teams, ultimately reinforcing the study’s primary goal of integrating manual expertise and machine learning models to help with better maintenance strategies.

4.3. XGBoost results

The XGBoost classification report highlights the model’s precision, recall and F1 score for both classes (0 for non-failures and 1 for failures). The model achieves an accuracy of 0.76 for

class 1, indicating that 76% of predicted failures were accurate. The recall of 0.64 for class 1 suggests the model correctly identified 64% of actual failures. The F1-score for class 1 is 0.70, reflecting a reasonable balance between precision and recall. Overall accuracy is 98%, emphasising the model’s strength in correctly classifying the dominant class (0), which is critical for real-world reliability (Figure 10). The weighted average F1-score of 0.98 further reinforces the model’s performance across all classes. These metrics align with the research objective of enhancing predictive maintenance through machine learning, as they demonstrate the model’s capacity to detect failures with high precision while maintaining overall system reliability.

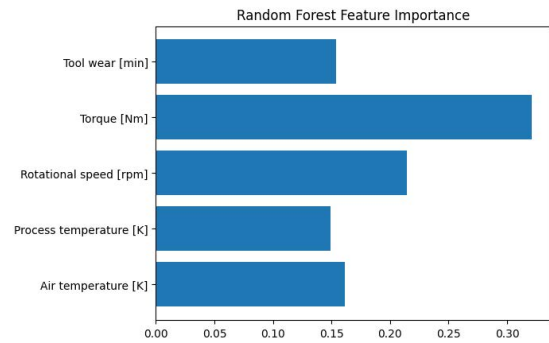


Figure 9: Random Forest Feature Importance.

```

XGBoost Classification Report:
      precision    recall  f1-score   support

0         0.99      0.99      0.99     1939
1         0.76      0.64      0.70      61

accuracy          0.98     2000
macro avg         0.88      0.82      0.84     2000
weighted avg      0.98      0.98      0.98     2000
    
```

Figure 10: Classification Report of XGBoost.

The XGBoost confusion matrix details the model’s predictions. True negatives (1,927 of 1,939 non-failure cases) and true positives (39 of 61 failure cases) were correctly classified by the model. However, 12 were incorrectly as failures (false positives) and 22 failed incorrectly as non-failures (false negatives) (Figure 11). The study finds moderate improvement in failure detection while the model continues to perform extremely robustly at predicting non-failures. Fewer false positives prove that the model was added correctly and was not repaired where needed. Fewer false negatives are necessary to remove mistaken repair. The confusion matrix is consistent with integrating advanced algorithms such as XGBoost to develop a robust predictive maintenance system.

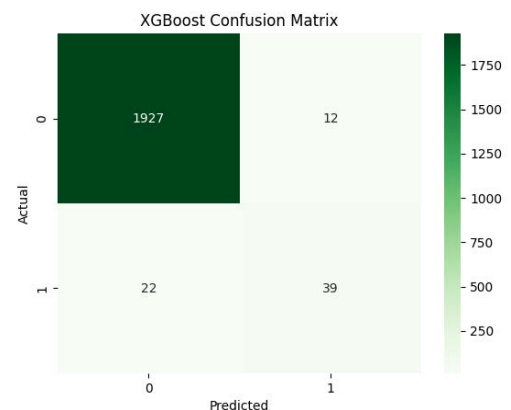


Figure 11: Confusion Matrix XGBoost.

ROC curve means the tradeoff between true positive rate (sensitivity) and false positive rate at different thresholds. The strong tendency of the model to discriminate classes is suggested by the curve sitting near the upper left corner and the steep beginning rise. Further quantification of this ability is provided by the area under the ROC curve (AUC) with values of approximately one corresponding to excellent classification performance (Figure 12). With this high AUC, we validate our objective of integrating machine learning into predictive maintenance systems by demonstrating a model’s capability to produce reliable and actionable predictions. The model also facilitates its practical and scalable real-time implementation in automated maintenance workflows, closing the research gap in practical and scalable predictive systems.

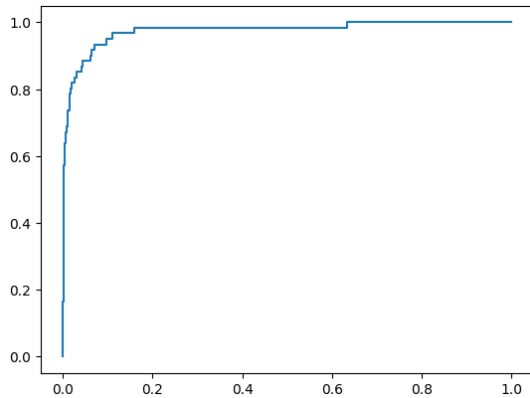


Figure 12: ROC Curve XGBoost.

The feature importance chart for XGBoost has very similar results to what Random Forest found, with Torque [Nm] at the top and then Rotational Speed [rpm] and Tool Wear [min]. It displays the robustness of these parameters to predict failures: this consistency across models. It also emphasises the importance of Air Temperature [K] and Process Temperature [K] to mechanical operations. This supports using machine learning alongside domain expert knowledge to improve predictive maintenance. The second focus of the research aim is the capability of the XGBoost to deal with feature interactions, which further increases its appeal in real-time industrial systems (Figure 13).

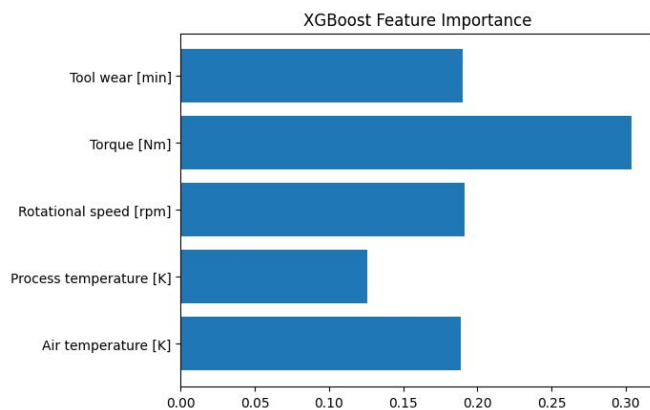


Figure 13: XGBoost Feature Importance.

4.4. Comparison

The ROC curve comparison highlights the discriminatory performance of Random Forest and XGBoost models. XGBoost achieves a slightly higher AUC (0.98) than Random Forest (0.97), indicating its superior ability to balance accurate positive and false favourable rates. Both curves demonstrate strong

predictive capabilities, reflecting their reliability in identifying equipment failures and non-failures. The near-identical performance aligns with the research objective of evaluating advanced machine learning models for predictive maintenance (Figure 14). The slight edge of XGBoost suggests it may be more effective for imbalanced datasets, a critical consideration in industrial predictive systems aiming for reduced false negatives and operational downtimes.

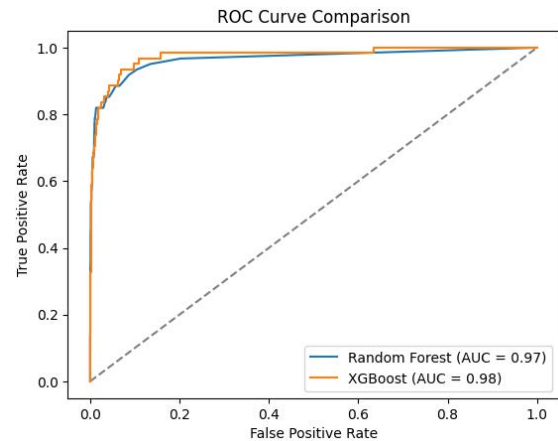


Figure 14: ROC Curve Comparison.

5. Discussion

5.1. Interpretation of results

This study presents the results that show the importance of features like Torque [Nm], Tool Wear [min] and Rotational Speed [rpm] in predictive maintenance. In both Random Forest and XGBoost, these features have always remained the most influential. For instance, Torque [Nm], an indication of the mechanical stress on the equipment, was the dominant predictor of machine failures. Tool Wear [min] also showed how gradual degradation over time was appropriate for determining maintenance needs when a critical failure is imminent.

The findings are closely aligned with domain expertise, which identifies torque and wear as leading indicators of equipment health. Validation from the models’ ability to quantify the relative importance of these features validates their significance and serves as actionable inputs for maintenance teams. By placing these variables in priority order, decision-makers can devote efforts to monitoring and controlling the most critical factors of machine performance, consequently lowering downtime and operating efficiency.

5.2. Implications and contributions

This research has implications in industrial settings where predictive maintenance can dramatically alter operations. This study also offers one of the most significant contributions to reducing unplanned downtime. Accurate failure prediction using critical features lets you schedule proactive maintenance activities, relieving disruptions in budget. It directly translates into reduced operational costs, as resources are closely coupled and we can reduce the probability of catastrophic failures.

The enhanced feature importance analysis also extensively contributes to other key ones, such as improved decision-making. With the insights modelling approaches such as Random Forest and XGBoost provide, maintenance teams can pinpoint and priorities the most critical variations in driving failures. Not only do these improve the accuracy of the prediction, but they

allow teams to make data-based decisions. It can help target maintenance scheduling and utilization of resources by focusing on torque and tool wear. Moreover, coupling machine learning with manual learning bridges the entrance between legacy and more innovative technological options, providing a sensible and scalable structure for predictive upkeep.

5.3. Challenges and limitations

There are challenges and limitations of the study, though it nonetheless contributes. The imbalance in data was one of the main problems to grapple with and the number of failure cases was far fewer than that of non-failure cases. It also made the models unable to reach high recall for predicting failure, as evidenced by the lower recall values for the failure class. Although weighted metrics and other more sophisticated algorithms like XGBoost mitigated some of this problem, more refining needs to be done to support balanced performance across all the classes involved.

However, there is another limitation of static data dependence on the models. The dataset helped give some insights, but it was without industrial environments' dynamic, real-time nature. This study does not consider the types of data streams that predictive maintenance systems often operate on, namely continuous data streams. Furthermore, although Random Forest and XGBoost worked fine, more advanced neural networks such as LSTM or CNN might provide better models for dealing with the temporal or spatial patterns in the data.

5.4. Future directions

Future research areas will integrate predictive maintenance systems and the Internet of Things (IoT) with cloud-based platforms. Continuous, real-time data streams offered by IoT-enabled sensors provide room for a more dynamic and responsive predictive maintenance framework. This should allow the system to scale out and let organisations keep track simultaneously from different locations. Testing and validating the models with real-time streaming data is another promising direction. The first is to build end-to-end pipelines from data collection, preprocessing and prediction in a single continuous flow. If the models are to be applied to industrial scenarios where timely interventions are necessary, then real-time data processing should be considered.

Finally, some of the limitations of this study can be addressed by exploring the use of advanced machine learning techniques like, for example, the deep learning models. Due to the popularity of neural networks like LSTMs for time-series data, these neural networks are desirable for modelling temporal patterns in real-time sensor data. Future work should also explore the integration of explainable AI (XAI) frameworks for better interpretability to enable maintenance teams to trust and act on model outputs with more confidence.

6. Automation and Deployment

For translating machine learning models to the practice of industrial applications such as predictive maintenance, effective automation and deployment of such systems is critical. The model deployment pipeline, integration of Java-based graphical user interfaces (GUIs) and a proposed workflow for automation by task schedulers or APIs are discussed in this section. The objective is to develop a scalable real-time system to solve downtime and operational inefficiency problems.

6.1. Model deployment pipeline

Saving the trained machine learning models for future use becomes the first step in deploying the predictive maintenance system. Random Forest and XGBoost models were saved using Python's joblib library. This approach also maintains all learned parameters and structure in serialisation, allowing for an uneventful model reload during deployment. They are saved as .pkl files so that they can be effortlessly integrated into other systems without the model retraining, which cuts the time spent on prediction by a significant margin.

These models are saved and reloaded using a Python-based API framework like Flask or FastAPI. These frameworks serve as a way to expose the models as endpoints so that other applications or systems can make real-time predictions by sending requests to the API. This flexibility is critically needed to facilitate interoperability with other industrial systems & tools.

6.2. Developing java-based GUIs for real-time predictions

A Java-based GUI was developed to combine with the predictive maintenance system to improve user interaction and accessibility. An interface called GUI is made by which the maintenance teams can input their new sensor data, see the prediction results and monitor the status of the equipment in real-time. UI was designed to visualize the predictions and the visualizations, e.g. failure probability and feature importance charts and leverages Java Swing or JavaFX. That integration closes the gap between our machine learning models and end users, giving non-technical staff access to advanced predictions. For example, if the torque, rotational speed or tool wear sensor readings are entered into the GUI, the system communicates with the Python API to find the prediction results. Then, presented in this simple form, these results allow maintenance teams to base their decisions on fundamentals without machine learning expertise.

7. Conclusion

This study focused on developing and evaluating predictive maintenance models using domain-specific knowledge and advanced machine learning algorithms to minimize equipment downtime and enhance operational efficiency. Through the AI4I 2020 dataset, the study assessed the propensity of two popular machine learning models, Random Forest and XGBoost, for failure prediction. Here, we showed that, in general, the accuracy of both models was high, with XGBoost being a bit better at dealing with imbalanced data and overall classification performance. Features such as Torque [Nm] and Tool Wear [min] are highlighted as the main predictors of equipment failures. Across these models, these variables consistently emerged as the most influential, consistent with domain expertise, which indicates mechanical stress being equated with torque and tool wear being an actual measure of equipment degradation. The study's ability to identify and rank such critical parameters bolsters its relevance in bridging the gap between advanced analytics and implementation. It contributes to creating actionable insights for maintenance teams.

The main contribution of this study is the comparison framework it established for earlier evaluation of machine learning models for predictive maintenance. By systematically comparing Random Forest and XGBoost and highlighting their

strengths and weaknesses, the research also offers considerable guidance in selecting the correct algorithm based on particular industrial demands. For example, while Random Forest has great interpretability and ease of use, XGBoost achieved better performance in predicting minority failure cases, making it more appropriate for applications with high precision for failure identifications.

Additionally, the study dealt with the practical issues of deployment of predictive maintenance systems in an industrial environment. Saving and serving models with APIs, integrating them with Java-based GUI for real-time predictions and automating workflows with task schedulers were detailed. These contributions serve as a base for the design and deployment of scalable, user-friendly, predictive maintenance systems that leverage the power of machine learning yet deliver an intuitive and robust interface.

Therefore, this paper built upon its goal of leveraging machine learning to advance predictive maintenance capabilities. It also produced a more practical framework for integrating these models into a real-world system. The insights gained from the work presented in this thesis have substantial potential for reducing downtime, enhancing maintenance schedules and reducing operational inefficiency. With this foundation, future work could extend this further to build predictive maintenance solutions with real-time data streams, advanced deep learning techniques and IoT-enabled systems to improve scalability and accuracy. The findings presented in this study are crucial for industries looking to approach change from reactive to proactive maintenance with data-driven insights maximizing their capabilities in decision-making and resource management.

8. Declarations

8.1. Funding: None.

8.2. Availability of data and materials: Available on request.

8.3. Authors' contributions: The author contributed equally to the execution of the research and write-up of this manuscript.

8.4. Ethics approval and consent to participate: Not needed.

8.5. Patient consent for publication: Not needed.

8.6. Competing interests: The author declares no conflict of interest.

9. References

- Sharanya S. A cyber physical system framework for industrial predictive maintenance using machine learning, in *Real-Time Applications of Machine Learning in Cyber-Physical Systems*, 2022: 241-269.
- Welz Z. Integrating Disparate Nuclear Data Sources for Improved Predictive Maintenance Modeling: Maintenance-Based Prognostics for Long-Term Equipment Operation, 2017.
- Kamath DU and Choppella K. *Mastering Java Machine Learning*. Birmingham: Packt Publishing, 2017.
- Python PDAU and Swamynathan M. *Mastering Machine Learning with Python in Six Steps*.
- Dinov ID. *Data science and predictive analytics*. Cham, Switzerland, 2018.
- Afridi YS, Ahmad K and Hassan L. Artificial intelligence based prognostic maintenance of renewable energy systems: A review of techniques, challenges and future research directions. *International Journal of Energy Research*, 2022;46: 21619-21642.
- Pundir A, Maheshwari P and Prajapati P. Machine learning based predictive maintenance model. in *Proceedings of the 2nd Indian International Conference on Industrial Engineering and Operations Management*, 2022.
- Achouch M, et al. On predictive maintenance in industry 4.0: Overview, models and challenges. *Applied Sciences*, 2022;12: 8081.
- Ezeigweneme CA, et al. Smart grids in industrial paradigms: a review of progress, benefits and maintenance implications: analyzing the role of smart grids in predictive maintenance and the integration of renewable energy sources, along with their overall impact on the industri. *Engineering Science & Technology Journal*, 2024;5: 1-20.
- Abbas A. AI for predictive maintenance in industrial systems. *International Journal of Advanced Engineering Technologies and Innovations*, 2024;1: 31-51.
- Hector I and Panjanathan R. Predictive maintenance in Industry 4.0: a survey of planning models and machine learning techniques. *PeerJ Computer Science*, 2024;10: 2016.
- Mohammed NA, et al. Performance Analysis of Different Machine Learning Algorithms for Predictive Maintenance. *AI-Khwarizmi Engineering Journal* 2024;20: 26-38.
- Ahmed NS. Machine Learning Models for Pavement Structural Condition Prediction: A Comparative Study of Random Forest (RF) and eXtreme Gradient Boosting (XGBoost). *Open Journal of Civil Engineering*, 2024;14: 570-586.
- Gawde S, et al. Explainable Predictive Maintenance of Rotating Machines Using LIME, SHAP, PDP, ICE. *IEEE Access*, 2024;12: 29345-29361.
- Gami SJ and Jain SN. Integrating IoT Data Streams with Machine Learning for Predictive Maintenance in Industrial Systems. *International Journal of Sustainable Development Through AI, ML and IoT*, 2024;3: 1-16.
- Dahiya S. Developing AI-Powered Java Applications in the Cloud Harnessing Machine Learning for Innovative Solutions. *Innovative Computer Sciences Journal*, 2024;10.
- Qazi AA and Abbas E. Big Data and Java are integrated with machine learning. *International Journal of Multidisciplinary Sciences and Arts*, 2024;3: 289-297.
- Dahiya S. Harnessing Cloud Computing for Enterprise Solutions: Leveraging Java for Scalable, Reliable Cloud Architectures. *Integrated Journal of Science and Technology*, 2024;1.
- Rossetto AGdM, et al. Enhancing Monitoring Performance: A Microservices Approach to Monitoring with Spyware Techniques and Prediction Models. *Sensors*, 2024;24: 4212.
- Lekidis A, et al. Predictive Maintenance Framework for Fault Detection in Remote Terminal Units. *Forecasting*, 2024;6: 239-265.
- Ucar A, Karakose M and Kırımça N. Artificial intelligence for predictive maintenance applications: key components, trustworthiness and future trends. *Applied Sciences*, 2024;14: 898.
- Tursunaliyeva A, et al. Making Sense of Machine Learning: A Review of Interpretation Techniques and Their Applications. *Applied Sciences*, 2024;14: 496.
- Ben Brahim A and Limam M. Ensemble feature selection for high dimensional data: a new method and a comparative study. *Advances in Data Analysis and Classification*, 2018;12: 937-952.
- Shehadeh A, et al. Machine learning models for predicting the residual value of heavy construction equipment: An evaluation

of modified decision tree, LightGBM and XGBoost regression.
Automation in Construction, 2021;129: 103827.