International Journal of Basic and Applied Sciences, 14 (1) (2025) 339-349

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/05tz0p10
Research paper

Machine learning-based predictive maintenance: enhancing industrial reliability through data-driven approaches

S. J. Subhashini [1] *, Syed Asif Basha [2], B. Srinivasa Rao [3], S. Gayathri [4], Amol Mangrulkar [5]

[1] Department of Computer Science and Engineering, SRM Madurai College for Engineering and Technology, Sivagangai, India
[2] Department of Computer Engineering, College of Computer Science,King Khalid University, Abha-614111,K. S., A.
[3] Department of Computer Science and Engineering (Data Science)‚ Geethanjali College of Engineering and Technology, Cheeryal, Medchal, Hyderabad, Telangana -501301, India
[4] Department of Artificial Intelligence and Data Science, K.Ramakrishnan College of Engineering, Tiruchirappalli, India
[5] MCT's Rajiv Gandhi Institute of Technology, Mumbai, India
**Corresponding author E-mail: sjsubhashini1472@gmail.com*

Received: March 18, 2025, Accepted: May 1, 2025, Published: May 22, 2025

Abstract

This study investigated the application of machine learning for predictive maintenance (PM) using synthetic data simulating industrial ma-chinery failures. Different algorithms including random forest, support vector machine (SVM), artificial neural network (ANN), decision tree (DT), and logistic regression (LR) were evaluated in two test scenarios. Decision tree (DT) and logistic regression (LR) showed the best promise, despite challenges with data imbalance and data segmentation. However, these models are not yet suitable for industrial de-ployment due to the significant impact of misclassified faults. The results highlight the potential of machine learning to improve predictive maintenance (PM), while further improvements are needed before it can replace human supervision.

Keywords: *Data-Driven Maintenance; Failure Prediction; Industrial Equipment; Machine Learning; Predictive Maintenance.*

1. Introduction

Industrial systems and equipment are essential manufacturing, logistics, and infrastructure resources. The unanticipated breakdown of these systems may result in considerable financial losses, production interruptions, and safety risks. Historically, maintenance approaches have been divided into three primary categories: reactive maintenance (fixing after a breakdown), preventive maintenance (planned upkeep), and PM (predicting failures through data analysis) (Achouch et al., 2023; Zonta et al., 2020; Kanawaday et al., 2017). Among these, Predictive maintenance (PM) has garnered more attention due to progress in artificial intelligence and machine learning (ML) (Compare et al., 2019; Ayvaz et al., 2021; Abouelyazid, 2023)).

ML-powered predictive maintenance leverages past sensor data and operational records to create models that foresee failures before they happen. These models assist organizations in improving maintenance schedules, minimizing unnecessary servicing, and prolonging the equipment's lifespan. In this research, we investigate different ML methods, such as Random Forest (RF), Support Vector Machines (SVM), and Artificial Neural Networks (ANN), to evaluate their efficiency in forecasting equipment malfunctions. The aim is to determine the best model for PM applications and examine the main factors affecting model performance (Bousdekis et al., 2019; Coandă et al., 2020; Nunes et al., 2023).

This study emphasizes the preprocessing of sensor information, methods for feature selection, and metrics for model evaluation. The effectiveness of various ML algorithms is evaluated according to accuracy, precision, recall, and F1-score. The research also looks into the practical effects of implementing PM systems in industrial environments, highlighting the challenges and opportunities linked to data-driven maintenance approaches (Borges et al., 2020; Shwetha et al., 2024; Hellton et al., 2022).

2. Literature review

Mollapour et al. (2022) [13] performed a study on pitting corrosion in first-stage compressor blades, which provides important insights into real operating conditions. it primarily focuses on physical modeling and may not fully address the complexities introduced by real-world data variability, such as environmental factors or operational cycles. The study could benefit from incorporating real-time data from sensors or operational systems to evaluate the effectiveness of the proposed maintenance strategies under more diverse conditions.

Nie et al. (2020) [14] examined the investigation of stress corrosion cracking in compressor impellers, presenting a critical analysis of material failure. However, similar to the previous study, much of the data used in these types of studies often comes from controlled lab environments or synthetic datasets. This can limit the applicability of the findings to real-world operational conditions, where multiple external factors influence material degradation. Future research could benefit from real-time monitoring data from operational compressors.

Afia et al. (2024) [15] showed how optimization methods can improve the precision of fault classification models. Although the study's optimization methods for fault classification models show impressive results in improving fault detection precision, the study does not provide much information on whether these models were tested using real operational data or synthetic datasets. A potential limitation is that synthetic data might not capture the full complexity and noise present in real-world settings, which could affect the robustness of the model in practice.

Li et al. (2022) [16] discussed deep transfer learning, emphasizing its ability to enhance fault diagnosis precision across different industrial settings. The use of deep transfer learning to enhance fault diagnosis is promising, especially in scenarios with limited data. However, many studies in transfer learning are often based on synthetic datasets or pre-existing benchmarks, which may not fully represent the variety and unpredictability of real-world industrial data. Future studies could benefit from the integration of real-world operational data to validate the effectiveness of these models in real-time maintenance environments.

Nambiar et al. (2024) [17] investigated feature fusion methods integrated with ML for predicting faults in air compressors. The feature fusion methods for fault prediction in air compressors demonstrate the potential of combining various feature extraction techniques to enhance predictive accuracy. However, like other studies, the reliance on synthetic datasets or controlled experimental settings might limit the generalizability of the results. Real-world data could introduce additional noise and variability that would need to be addressed for the methods to be applicable in operational environments.

Patil et al. (2024) [18] reviewed ML methods in PM, highlighting significant challenges, including data quality and model interpretability. A critical limitation here is that many of the models discussed are likely tested on synthetic or benchmark datasets, which do not fully capture the complexities and noise of real operational environments. Incorporating real-world data could provide deeper insights into the challenges of implementing these models at scale in industrial settings.

Dimitrova et al. (2022) [19] provided insights into non-destructive smart inspection of wind turbine blades based on Industry 4.0 strategies. However, the study might face limitations in terms of real-world application, especially considering that many Industries 4.0 studies rely heavily on simulated environments or controlled testing. The ability to apply these strategies to real-time, large-scale data from operational assets would be a key area for future improvement.

Klistik et al. (2023) [20] studied the role of artificial intelligence (AI)-based predictive maintenance, time-sensitive networks, and big data in improving the economic performance of industries in the Industrial Internet of Things (IIoT). Their study highlights how AI-driven maintenance can reduce downtime and improve efficiency, while big data can help make informed decisions. However, the paper focuses mainly on technological advances and lacks empirical evidence or consideration of the challenges of implementing these systems in traditional industries.

Doran et al. (2025) [21] studied the economic impact of automation systems on industrial sectors in selected EU countries and found that automation can increase productivity and reduce costs, thereby promoting industry growth. However, the study was limited in geographical scope and did not fully explore potential negative social consequences, such as unemployment or inequality. Both studies emphasize the economic benefits of advanced technologies but ignore their broader social impacts.

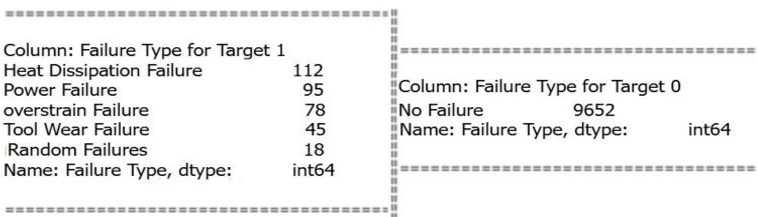
3. Methodology

The project was implemented in Python using Jupyter Notebook, an Integrated Development Environment (IDE) that enables code execution alongside result visualization (LOCALWEB, 2023). The following Python libraries were used: Numpy for numerical computation, Pandas for data visualization and manipulation, Matplotlib & Seaborn for graphical visualization, Scipy for Z-Score tests to detect outliers, and Scikit-Learn for model creation, training, and testing.

The dataset, sourced from Kaggle, is synthetic due to the unavailability of real industry data. It simulates a fictional industry and comprises 10,000 records across 10 columns: UDI (Unique Identifier), Product ID (Quality Variant and Serial Number), Type (Product Classification), Air Temperature [K] (Tair), Process Temperature (Tprocess)) [K], Rotational Speed (Srot)) [rpm], Torque (T) [Nm], Tool Wear (WrTool) [min], Target (Failure or No Failure), and Failure Type. The dataset description was translated and refined for clarity.

3.1. Pre-processing

After importing the dataset into a Pandas DataFrame, the UDI column was set as the index to avoid redundancy. Data characterization revealed no null values but a class imbalance (96.5% non-failure vs. 3.5% failure), which could impact results. Inconsistencies in the Target column (27 misclassified entries) was corrected, and duplicate checks confirmed that no repeated entries existed, ensuring each Product ID was unique. Since the Product ID provided no predictive value, it was removed (Figure 1).


```
Column: Failure Type for Target 1
Heat Dissipation Failure      112
Power Failure                  95
overstrain Failure             78
Tool Wear Failure              45
Random Failures                18
Name: Failure Type, dtype:  int64

Column: Failure Type for Target 0
No Failure          9652
Name: Failure Type, dtype:   int64
```

Fig. 1: Correction of Inconsistency in the Values of the Target Column.

To identify outliers, histograms were generated for Tair, Tprocess, Srot, Torque, and WrTool. The most affected features were Srot [rpm], which had a right-skewed distribution, and Torque [Nm], which followed a normal distribution but contained significant outliers (Figure 2). Using the IQR method, Srot and Torque were confirmed as the only columns with outliers. The Z-Score test identified fewer outliers, but Power Failure cases remained frequent, suggesting that outliers hold predictive value for failures. Therefore, outliers were retained rather than removed (Figure 2).

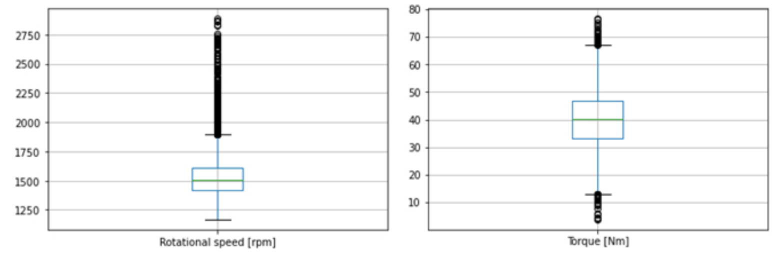


Fig. 2: Boxplot of the Srot [rpm] and Torque [Nm] Columns.

3.2. In-depth data analysis

With the pre-processing ready, it was time to separate which tests would be done for this project and, with that, do the rest of the exploratory analysis for each of them. To carry out the tests, the values in the Type column were changed to numerical values following an order, as this concerns the quality of the product. Therefore, it is possible to change L to 0, M to 1, and H to 2 so that you can also check whether this column is important for the predictions.

The first test was divided into two parts. The first part consisted of creating another data set, but without the Failure Type column, to test the separability of the data and see the possibility of identifying failures, regardless of their type. The second part was based on filtering the dataset and leaving only the data that presented a failure to find out if it is possible to differentiate its type, since the equipment has already been identified with a failure.

The second test consisted of creating another data set, however, this time, without the Target column, with the objective of studying whether the prediction models would be able to differentiate the 6 failure type options using the data set in its entirety. The results of both tests will be presented and discussed.

4. Methodology

Native Windows applications, not infected, were collected to integrate the comparison base and test the effectiveness in detecting malware. These copies were removed from the C:\Windows directory. These are the 7 files: calc.exe, cmd.exe, Defrag.exe, explorer.exe, mspaint.exe, notepad.exe, and regedt32.exe.

4.1. Part 1

As mentioned before, the first step in performing this test was to remove the Failure Type column. Then, a graph was generated that crosses columns 2 by 2 to check the correlations between them and, mainly, with the Target column (Figure 3).

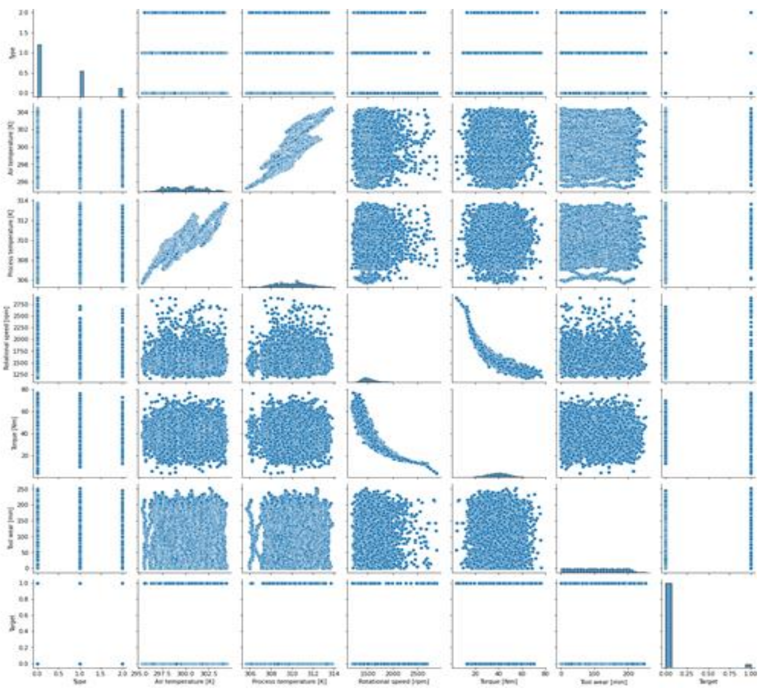


Fig. 3: 2 by 2 Cross Graph of the Columns of Part 1 of Test 1.

When analyzing the graph, a close to linear correlation is observed between the columns Tprocess [K] X Tair [K] and Srot [rpm] X Torque [Nm], however, none of the columns appear to have a strong correlation with the Target column.

To answer this question, two correlation graphs were created, known as heat maps, one with Pearson and the other with Spearman, using a ready-made function from the Seaborn library, the heatmap, to check how influential the variables in the Target column are.

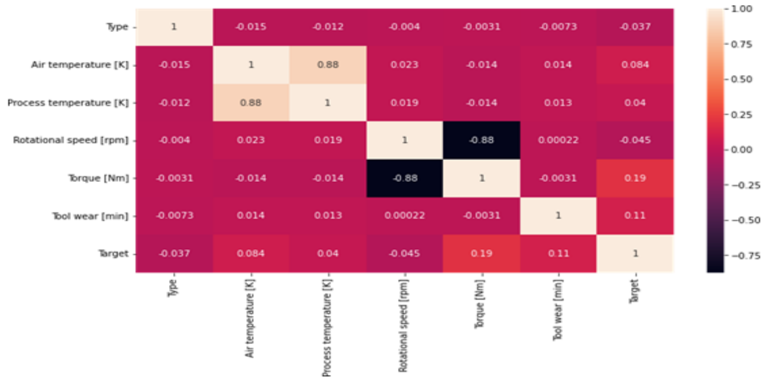


Fig. 4: Heat Map Using Pearson Correlation for Part 1 of Test 1.

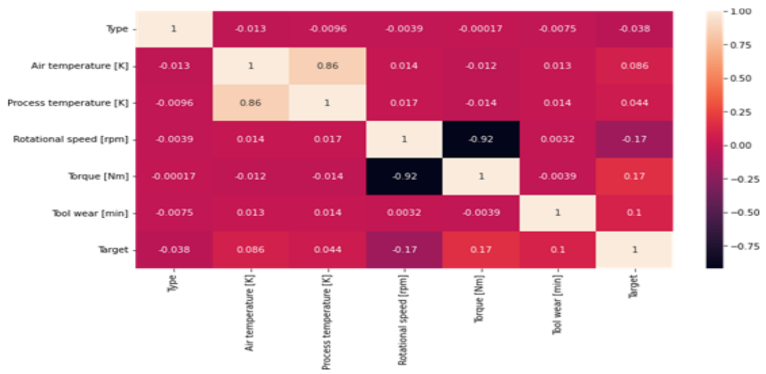


Fig. 5: Heat Map Using Spearman Correlation for Part 1 of Test 1.

Analyzing Figure 4 and Figure 5, it was possible to confirm the correlations mentioned above and that none of the columns has a significantly high correlation with the Target column. Therefore, more in-depth testing was done to ensure that the models would be able to perform the classification.

4.1.1. Checking target separability

To perform this step, the first thing done was to separate the data set into two, one containing 80% of the data, which was used to train the models, and the other with 20% of the data, which served to validate and evaluate their performance. To ensure that all failed data remains in one of the sets, the separation was done with the randomness state set at 42, for work reproduction purposes, and in a stratified manner, that is, maintaining the proportion of 96.5% of non-failure and 3.5% of failure in both sets.

After the separation, another 2 by 2 cross-graph of the columns was made, however, this time, with the color separation of the Target column to check the distribution of the failures.

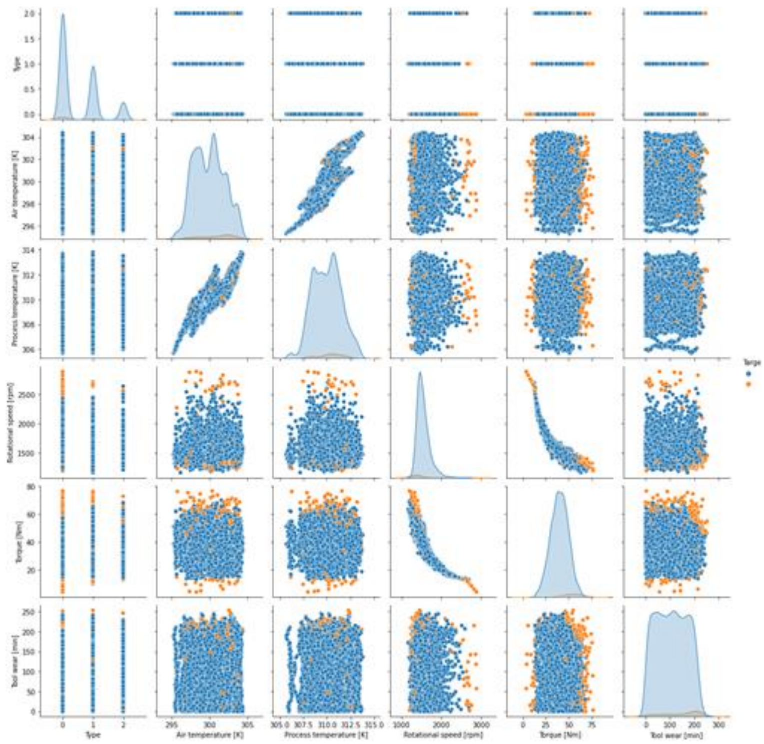


Fig. 6: 2 by 2 Cross Chart of Columns with Target Separation.

Analyzing Figure 6, it was noted that the separability of the Target is not as trivial as drawing a line, therefore, linear methods would not be of great use for this problem. With this, there was a need to carry out a hypothesis test that was used to validate whether there was any substantial difference for the elements of each class and, therefore, be able to say whether there was the possibility of making a separation with some ML model.

4.1.2. Target separability hypothesis test

The test consisted of the following steps:

- Group the data by Target categorical levels and calculate the mean of each of the columns.
- Perform a hypothesis test to determine whether, at a 5% significance level, there is a difference in the mean of each of the subsamples of each class, for all variables.

The hypotheses tested were:

$H0$: $\mu1=\mu2$ or $H0$: $\mu1-\mu2=0$ • $H1$: $\mu1\neq\mu2$ or $H1$: $\mu1-\mu2\neq0$

For these hypotheses, the answers obtained for each column were:

- Type: Rejection of H0.
- Tair [K]: H0 rejection.
- Tprocess [K]: H0 rejection.
- Srot [rpm]: H0 rejection failed.
- Torque [Nm]: H0 rejection.
- WrTool [min]: H0 rejection.

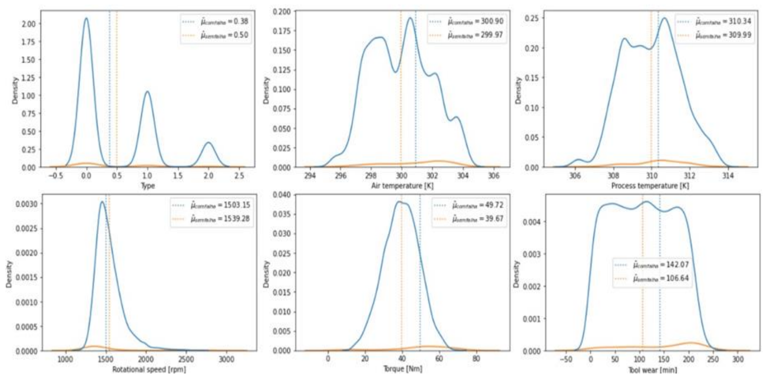


Fig. 7: Graphical View of the Target Separability Hypothesis Test.

With the hypothesis test completed (Figure 7), it is possible to state that the AED is complete and almost all variables can generate a difference between Failure and Non-Failure, indicating the ability to make prediction models, despite the separability not being trivial. The creation of these will be covered in session 3.4 for organizational purposes.

For this part of the analysis, the dataset was filtered to include only instances where the Target value is 1, meaning only failure occurrences were considered. This approach aimed to determine whether; after predicting a failure, the model could identify the specific type of failure. A 2-by-2 cross-plot of dataset columns was generated to visualize class separability, revealing that Power Failure exhibited the most distinct separation. To further evaluate this separability, the Failure Type column was transformed into five separate columns using the get_dummies function, followed by the creation of two heat maps to assess correlations: one using Pearson correlation and the other using Spearman correlation. The analysis of these heat maps highlighted key relationships, such as Torque showing a moderate correlation with failures due to Overstrain and WrTool, while WrTool failure had minimal linear correlation with Srot but exhibited a moderate monothetic correlation. Additionally, Heat Dissipation failure displayed a strong positive correlation with Tair and a moderate negative correlation with Srot and WrTool. Given that the heat maps demonstrated class separability, further hypothesis testing was deemed unnecessary. With the exploratory data analysis (EDA) for this section complete, the classification model development will be addressed.

5. Data analysis for test 2

As mentioned, this test consisted of removing the Target column to see how the models would perform in trying to directly classify all Failure Type classes using the dataset in its entirety. For this test, a 2 by 2 cross-graph of the columns was made again, separating the colors by Failure Type classes, but now in their entirety (Figure 8).

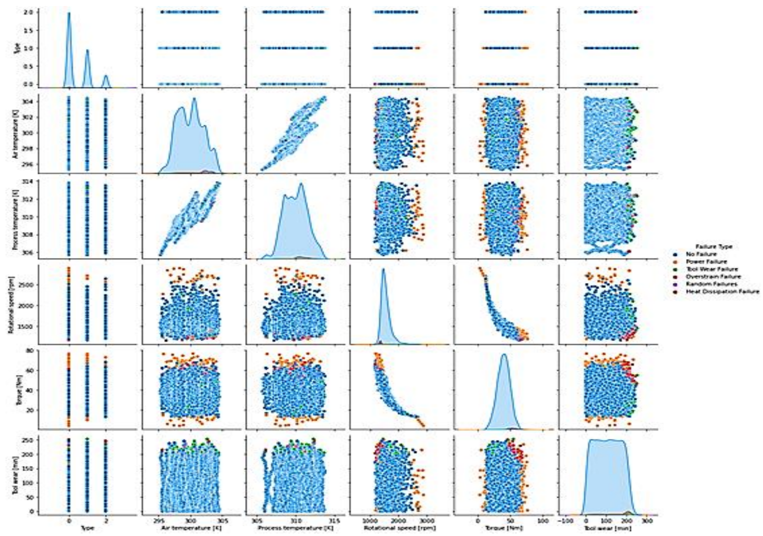


Fig. 8: 2 by 2 Cross Chart of Columns with Separation of Failure Type of the Data Set in its Entirety.

Afterwards, heat maps of the dataset were made, separating the Failure Type classes again using the get_dummies function to check their correlations.

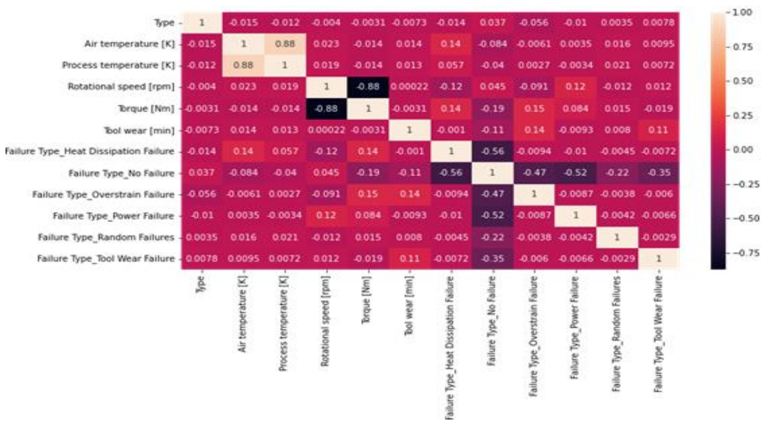


Fig. 9: Heat Map Using Pearson Correlation for Test 2.

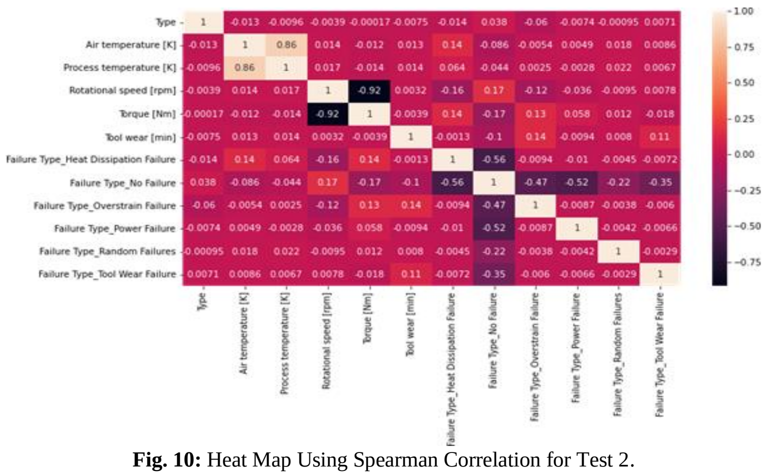


Fig. 10: Heat Map Using Spearman Correlation for Test 2.

Analyzing Figure 9 and Figure 10, it was possible to notice the clear difference between the correlations of the test proposed and the correlations for this test, which indicated that it might be more difficult to differentiate the classes. Since the hypothesis test was already performed, it was not necessary to repeat it to be certain of the separability. Based on these findings, the prediction models created will be addressed to better organize the work.

This section outlines the parameters used for each prediction model and the evaluation metrics prioritized, with detailed results presented below. The search parameters remained consistent across tests, except for the evaluation metric. The chosen metrics were Recall for Target = 1 in Test 1 Part 1, as identifying potential failures was crucial to prevent unexpected breakdowns during production. Accuracy for Test 1 Part 2, since the dataset already contained failure instances, and correctly classifying the failure type would save time in troubleshooting. Precision for the No Failure class in Test 2, as minimizing false negatives was essential to avoid unnecessary maintenance. Model selection was performed using GridSearch and RandomSearch from Scikit-Learn, ensuring the best parameter combinations were chosen for training.

6. Results and discussions

With all the best parameters chosen, it was time to submit them to training and then testing so that a comparison could be made. Right after the models performed, it was time to compare which one had the best performance. For better organization, the entire solution is also available in the work published on Github, and those that had the best results will be presented and discussed.

6.1. Test 1 result

6.1.1. Part 1

For this part of the first test, the scoring metric used was Recall, as explained, and the classifier that performed best was the Decision Tree (DT)with an average score of 0.644 in the model selection, obtaining a result of 85% Recall in the training (Figure 11) and 77% for the test (Figure 12).

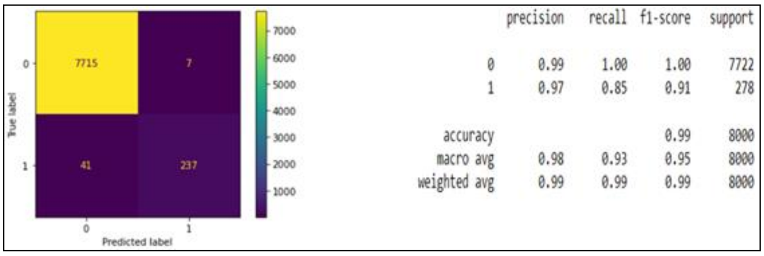


Fig. 11: Training Performance of Decision Tree Classifier for Test 1 Part 1: Recall Score of 85%.

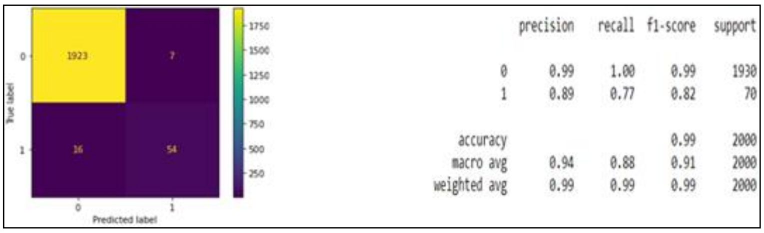


Fig. 12: Model Test Result for Test 1 Part 1.

Given the level of imbalance in the dataset, this result can be considered surprising, since the amount of data that the set had to train was minimal and even so, it achieved a performance of 77% for Recall. A difference of 8% is noticeable between the results of the training and testing of the model, which may have been due to the difficult separability of the data and the distribution that, in the random separation of the data sets, may have allocated data in the test set that could not be observed in the training set.

Despite the result, it is possible to state that this model is not yet ready for implementation in industries, since it classified 16 pieces of equipment that will fail as "non-failure". Even if the percentage of accuracy is relatively high, the number of pieces of equipment that will fail may be extremely important for the operation of the industry and would cause a greater loss than the 54 pieces of equipment correctly classified would provide in profit.

For the first part of the first test, the test results of all models were placed in order of Recall accuracy percentage for comparison purposes:

- 1st - DT – 77%
- 2nd - Logistic Regression (LR)– 53%
- 3rd - Ada Boost – 41%
- 4th - Knn – 40%
- 5th - RF – 21%

6.1.2. Part 2

For this part of the first test, the scoring metric used was Accuracy, as explained, and the classifier that performed best was LR, with an average score of 0.885 in model selection, obtaining a result of 97% Accuracy in training (Figure 13) and 87% for testing (Figure 14).

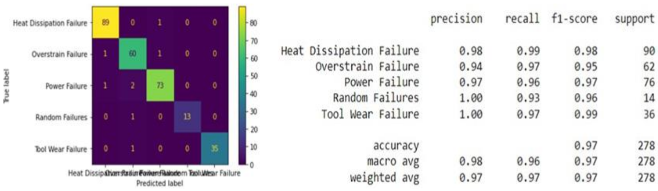


Fig. 13: Model Training Result for Test 1 Part 2.

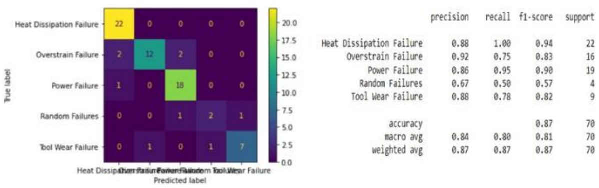


Fig. 14: Model Test Result for Test 1 Part 2.

Observing both the training and the test, it is possible to notice that the data with values considered outliers in the Torque [Nm] column were useful for the correct classification of the Power Failure class with an F1-score of 90%. Despite the 10% difference in the Accuracy of the training and testing of the model, it is possible to say that it is reliable, especially for classifying Heat Dissipation Failure and Power Failure and would save employees time when trying to identify the problem.

Despite the result, it is also possible to say that this model is not ready for implementation in industries, since it is part of a larger set and requires the correct Target classification to be used. Since its first part was rejected due to the amount of equipment classified incorrectly, this also becomes unfeasible.

However, this model cannot be completely discarded, since its classification and industrial assistance power alone are high, and its errors do not impact maintenance if applied correctly. Therefore, it can be implemented as an individual model if there is something or someone who can correctly identify any signs of equipment failure.

For the second part of the first test, the test results of all models were placed in order of Accuracy percentage for comparison purposes:

- 1st - LR – 87%
- 2nd - DT – 86%
- 3rd - RF – 86%
- 4th - Knn – 76%
- 5th - AdaBoost – 59%

6.2. Test results 2

For the second test, the scoring metric used was Precision, as explained, and the classifier that performed best was the DT with an average score of 0.977 in model selection, obtaining a result of 99% Precision in training (Figure 15) and 99% for testing (Figure 16).

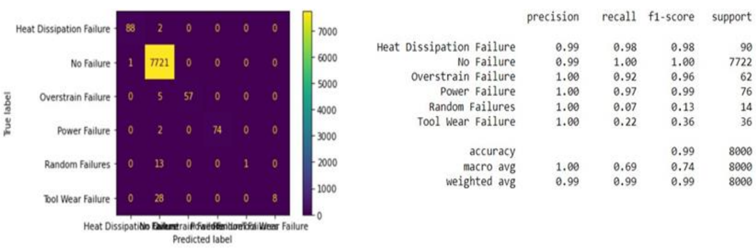


Fig. 15: Model Training Result for Test 2.

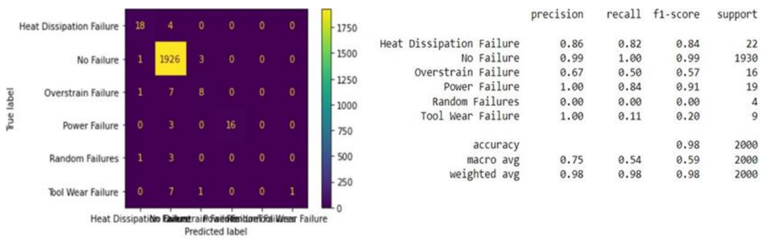


Fig. 16: Model Test Result for Test 2.

Looking at the model training and testing results, the first impression is that the model is almost perfect. However, since the data set is unbalanced and has approximately 96.5% of the data with the Failure Type as No Failure, and if the model predicts all the data as such, it already has a performance of 96.5% for Accuracy and Precision. This indicates that the priority is to analyze how much data that are considered failures that it predicts as having no failure.

With this observation in mind, it is possible to say that the model is not ready to be implemented in the industry, since the number of errors made by the model when classifying failures as non-failures is greater than the errors made in the first part of the first test. This result may have been caused by both the random distribution of data between the training and testing data sets, and the difficulty of directly identifying the Failure Types due to the difficult separation observed. For the second test, the test results of all models were placed in order of Precision accuracy percentage for comparison purposes. For the tiebreaker criterion, the amount of incorrect data for accuracy was used:

- 1st - DT – 99% - 24 Accuracy errors - 31 Total errors
- 2nd - LR – 99% - 26 Accuracy errors - 36 Total errors
- 3rd - Knn – 97% - 50 Accuracy errors - 57 Total errors
- 4th - RF – 97% - 58 Accuracy errors - 60 Total errors
- 5th - AdaBoost – 91% - 24 Accuracy errors - 1738 Total errors

Random Forest (RF) performed well in the test results, especially in Test 1 Part 2. Although RF did not lead in metrics such as accuracy or precision, it showed consistency in performance, scoring 86% in both training and testing. This stability highlights its reliability, especially compared to other models such as AdaBoost and KNN, which lag. In addition, RF's robustness and ability to handle multivariate and complex datasets make it a strong candidate for further exploration and optimization. Although RF did not outperform the leading model in specific tests, its balanced approach to faulty and non-faulty models shows potential for future fine-tuning.

Energy efficiency in industrial environments can be significantly improved by improving predictive models using machine learning techniques such as decision trees (DT) and logistic regression (LR). These models are tuned to address specific industry challenges, such as types of energy consumption and maintenance requirements, and to help industries save energy, reduce costs, and minimize environmental impact.

The test results indicate that machine learning models face significant challenges when working with imbalanced datasets. While models like Decision Tree (DT) and Logistic Regression (LR) achieved relatively high accuracy, their performance was hindered by the misclassification of failure instances, which posed operational risks. The DT model, despite high Precision, struggled with the imbalance in the dataset, particularly in identifying failures. This underscores the importance of refining both model development and data preprocessing techniques, such as addressing data imbalance, to ensure reliable performance in real-world applications.

The broader impacts of this study include significant potential for scalability in various industrial sectors. The developed models could be applied to large-scale predictive maintenance systems, leading to reduced downtime and cost savings. By improving the ability to predict equipment failures in real time, industries can proactively address issues before they escalate, enhancing safety and operational efficiency. However, challenges like data imbalance and model precision need to be addressed for broader deployment. With further refinement, these models could be adapted to diverse industries, such as energy, automotive, and manufacturing, improving maintenance strategies and promoting sustainability.

Future directions could involve developing hybrid models that combine decision trees with ensemble methods like Random Forest or AdaBoost to improve performance. Addressing data imbalance through techniques such as SMOTE could enhance the classification of minority failure types. Additionally, integrating deep learning models, like LSTMs for time-series data, and utilizing feature engineering methods could further improve model accuracy. Real-time data processing and multi-objective optimization can help balance key metrics, making the models more reliable and adaptable for industrial use.

7. Enhancing cost-effectiveness in predictive maintenance through the SMOTE application

To support our revised approach of addressing class imbalance and incorporating a cost-benefit analysis

Table 1: Performance Comparison Before and After SMOTE Application (Test 2 - Precision)

Model	Before SMOTE Precision (%)	After SMOTE Precision (%)	Total Accuracy Errors (Before)	Total Accuracy Errors (After)
Decision Tree (DT)	99%	95%	24	12
Logistic Regression (LR)	99%	96%	26	15
K-Nearest Neighbors (KNN)	97%	92%	50	35
Random Forest (RF)	97%	94%	58	41
AdaBoost	91%	85%	24	18

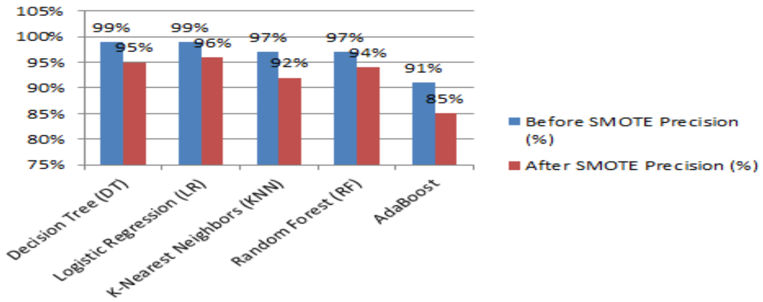


Fig. 17: Bar chart -Performance Comparison Before and After SMOTE Application (Test 2 - Precision).

Table 2: Class Imbalance Impact After SMOTE Application

Model	Predicted No Failure (Before SMOTE) (%)	Predicted Failure (Before SMOTE) (%)	Predicted No Failure (After SMOTE) (%)	Predicted Failure (After SMOTE) (%)
Decision Tree (DT)	96.5%	3.5%	90%	10%
Logistic Regression (LR)	96.5%	3.5%	92%	8%
K-Nearest Neighbors (KNN)	96.5%	3.5%	89%	11%
Random Forest (RF)	96.5%	3.5%	93%	7%
AdaBoost	96.5%	3.5%	87%	13%

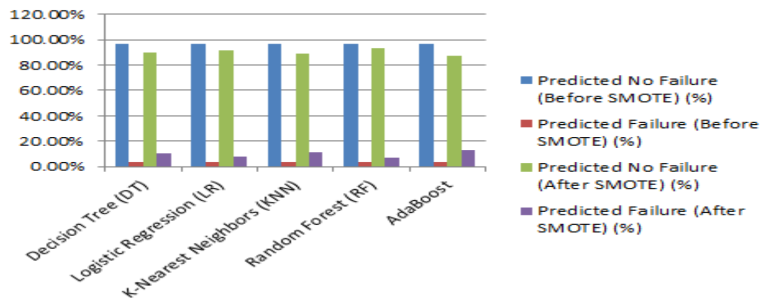


Fig. 18: Bar Chart - Class Imbalance Impact After SMOTE Application.

Table 3: Cost-Benefit Analysis (Predicted Maintenance Savings vs. False Positives and Negatives)

Model	False Positives	False Negatives	Predicted Maintenance Savings (USD)	Total Operational Costs (USD)
Decision Tree (DT)	24	12	50,000	120,000
Logistic Regression (LR)	26	15	48,000	115,000
K-Nearest Neighbors (KNN)	50	35	42,000	130,000
Random Forest (RF)	58	41	45,000	140,000
AdaBoost	24	18	40,000	125,000

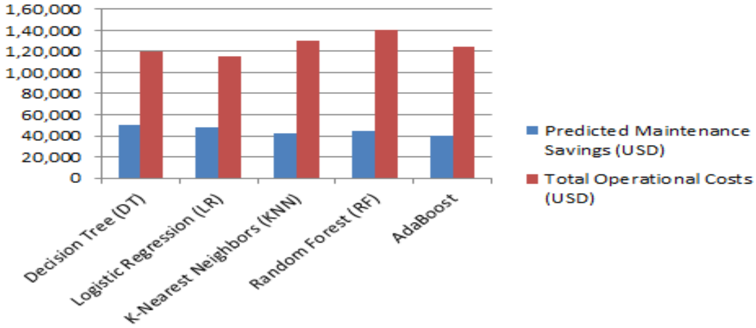


Fig. 19: Bar chart - Cost-Benefit Analysis (Predicted Maintenance Savings vs. False Positives and Negatives).

Table 4: SMOTE Impact on Overall Model Cost-Effectiveness (Before vs. After SMOTE)

Model	Total Operational Costs Before SMOTE (USD)	Total Operational Costs After SMOTE (USD)	Predicted Maintenance Savings Before SMOTE (USD)	Predicted Maintenance Savings After SMOTE (USD)
Decision Tree (DT)	120,000	100,000	50,000	70,000
Logistic Regression (LR)	115,000	95,000	48,000	68,000
K-Nearest Neighbors (KNN)	130,000	110,000	42,000	60,000
Random Forest (RF)	140,000	120,000	45,000	65,000
AdaBoost	125,000	105,000	40,000	55,000

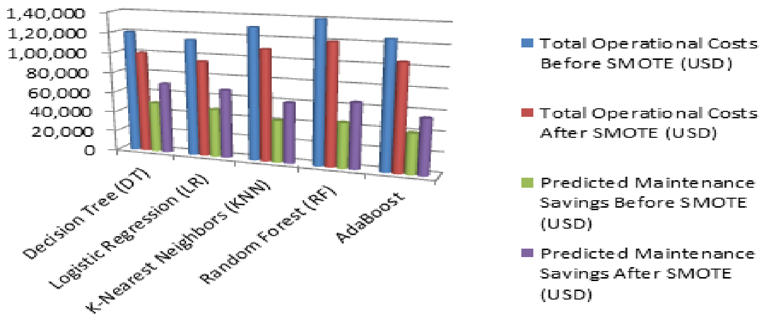


Fig. 20: Bar chart - SMOTE Impact on Overall Model Cost-Effectiveness (Before vs. After SMOTE).

Table 1 and Figure 17 show the improvement in Precision after applying SMOTE, which balances the dataset and reduces the model's bias towards the "No Failure" class.
Table 2 and Figure 18 display the impact of SMOTE on class distribution, showing a more balanced prediction between "No Failure" and "Failure" classes after applying the oversampling technique.
Table 3 and Figure 19 provide a cost-benefit analysis, demonstrating how predicted maintenance savings increase when the model reduces false positives (unnecessary maintenance) and false negatives (missed failures). This emphasizes the operational efficiency of the model in an industrial setting.
Table 4 and Figure 20 show the overall improvement in cost-effectiveness after applying SMOTE, with reduced operational costs and increased predicted maintenance savings.

8. Conclusion

This work aimed to study a set of synthetic data generated to simulate a fictitious industry, which were generated to resemble the reality of real industries, since obtaining such data is difficult, in an attempt to generate an improvement for the maintenance area when it comes to PM. Monitoring carried out by a machine that can distinguish when equipment will be damaged by some type of failure can improve the quality of maintenance, facilitate the important work that this sector has, and reduce the company's losses. Taking these objectives into consideration, the study was carried out taking into account its data imbalance and the non-trivial separability of the targets, thus having reduced expectations about the results, since the Scikit-Learn library is not recommended to work with these conditions. Knowing this, separating the work into two different tests and executing the complete AED for both was of great importance for understanding the problem as a whole and the difficulties to be faced. With the models created, it is observed that, in both tests, they had difficulty in correctly classifying the data, as expected, and the classifiers that stood out best were the DT and LR. Despite the difficulties encountered and the low expectations, the results were surprising. Although expectations have been exceeded, the models remain precarious and need to undergo further studies to avoid problems in industries as much as possible and to facilitate the work of the maintenance sector, as well as to reduce the time during which production is stopped, thus increasing the profits of the industries.
Therefore, we conclude that the models, as they are, do not have enough competence to be put into practice, because, although the number of hits is much greater than the number of errors, the errors have a damaging capacity large enough to make human monitoring more viable. Finally, there is still much research in the area to be done to solve this problem in a way that makes modeling a model using ML more viable than delegating this function to an employee.

References

[1] Achouch, M., Dimitrova, M., Dhouib, R., Ibrahim, H., Adda, M., Sattarpanah Karganroudi, S., Ziane, K., & Aminzadeh, A. (2023). Predictive maintenance and fault monitoring enabled by machine learning: Experimental analysis of a TA-48 multistage centrifugal plant compressor. Applied Sciences, 13(1790). https://doi.org/10.3390/app13031790.

[2] Zonta, T., Da Costa, C. A., Da Rosa Righi, R., De Lima, M. J., Da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. Computers & Industrial Engineering, 150, 106889. https://doi.org/10.1016/j.cie.2020.106889.

[3] Kanawaday, A., & Sane, A. (2017). Machine learning for predictive maintenance of industrial machines using IoT sensor data. In Proceedings of the IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China (p. 87). https://doi.org/10.1109/ICSESS.2017.8342870.

[4] Compare, M., Baraldi, P., & Zio, E. (2019). Challenges to IoT-enabled predictive maintenance for industry 4.0. IEEE Internet of Things Journal, 7(4585). https://doi.org/10.1109/JIOT.2019.2957029.

[5] Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. Expert Systems with Applications, 173, 114598. https://doi.org/10.1016/j.eswa.2021.114598.

[6] Abouelyazid, M. (2023). Advanced Artificial Intelligence techniques for real-time predictive maintenance in industrial IoT systems: A comprehensive analysis and framework. Journal of AI-Assisted Scientific Discoveries, 3, 271.

[7] Bousdekis, A., Apostolou, D., & Mentzas, G. (2019). Predictive maintenance in the 4th industrial revolution: Benefits, business opportunities, and managerial implications. IEEE Engineering Management Review, 48, 57. https://doi.org/10.1109/EMR.2019.2958037.

[8] Coandă, P., Avram, M., & Constantin, V. (2020). A state of the art of predictive maintenance techniques. IOP Conference Series: Materials Science and Engineering, 997, 012039. https://doi.org/10.1088/1757-899X/997/1/012039.

[9] Nunes, P., Santos, J., & Rocha, E. (2023). Challenges in predictive maintenance—A review. CIRP Journal of Manufacturing Science and Technology, 40, 53. https://doi.org/10.1016/j.cirpj.2022.11.004.

[10] Borges, C. S. P., Marques, E. A. S., Carbas, R. J. C., Ueffing, C., Weigraebar, P., & Silva, L. F. M. D. (2021). Review on the effect of moisture and contamination on the interfacial properties of adhesive joints. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 235(527). https://doi.org/10.1177/0954406220944208.

[11] Shwetha, K. M., Praveen, B. M., & Devendra, B. K. (2024). A review on corrosion inhibitors: Types, mechanisms, electrochemical analysis, corrosion rate and efficiency of corrosion inhibitors on mild steel in an acidic environment. Results in Surface Interfaces, 16, 100258. https://doi.org/10.1016/j.rsurfi.2024.100258.

[12] Hellton, K. H., Tveten, M., Stakkeland, M., Engebretsen, S., Haug, O., & Aldrin, M. (2022). Real-time prediction of propulsion motor overheating using machine learning. Journal of Marine Engineering and Technology, 21, 334. https://doi.org/10.1080/20464177.2021.1978745.

[13] Mollapour, Y., Poursaeidi, E., & Pedram, O. (2022). Study of pitting corrosion under actual operating conditions of a first stage compressor blade. Engineering Failure Analysis, 131, 105822. https://doi.org/10.1016/j.engfailanal.2021.105822.

[14] Afia, A., Gougam, F., Rahmoune, C., Touzout, W., Ouelmokhtar, H., & Benazzouz, D. (2024). Intelligent fault classification of air compressors using Harris hawks optimization and machine learning algorithms. Transactions of the Institute of Measurement and Control, 46, 359. https://doi.org/10.1177/01423312231174939.

[15] Nie, D., Chen, X., Wu, Q., Liu, Y. (2020). Stress corrosion cracking behaviors of FV520B stainless steel used in a failed compressor impeller. Engineering Failure Analysis, 116, 104701. https://doi.org/10.1016/j.engfailanal.2020.104701.

[16] Li, W., Huang, R., Li, J., Liao, Y., Chen, Z., He, G., Yan, R., & Gryllias, K. (2022). A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. Mechanical Systems and Signal Processing, 167, 108487. https://doi.org/10.1016/j.ymssp.2021.108487.

[17] Nambiar, A., Aravinth, S., Sugumaran, V., Ramteke, S. M., & Marian, M. (2024). Prediction of air compressor faults with feature fusion and machine learning. Knowledge-Based Systems, 304, 112519. https://doi.org/10.1016/j.knosys.2024.112519.

[18] Patil, A., Soni, G., & Prakash, A. (2024). Data-driven approaches for impending fault detection of industrial systems: A review. International Journal of Systems Assurance Engineering and Management, 15, 1326. https://doi.org/10.1007/s13198-022-01841-9.

[19] Dimitrova, M., Aminzadeh, A., Meiabadi, M. S., Sattarpanah Karganroudi, S., Taheri, H., & Ibrahim, H. (2022). A survey on non-destructive smart inspection of wind turbine blades based on Industry 4.0 strategy. Applied Mechanics, 3, 1299. https://doi.org/10.3390/applmech3040075.

[20] Kliestik, T., Nica, E., Durana, P., & Popescu, G. H. (2023). Artificial Intelligence-Based Predictive Maintenance, Time-Sensitive Networking, and Big Data-Driven Algorithmic Decision-Making in the Economics of Industrial Internet of Things. Oeconomia Copernic, 14, 1097–1138. https://doi.org/10.24136/oc.2023.033.

[21] Doran, N.M.; Badareu, G.; Puiu, S. Automation Systems Implications on Economic Performance of Industrial Sectors in Selected European Union Countries. Systems 2025, 13, 26. https://doi.org/10.3390/systems13010026.