

Machine learning for predicting failures of electronic components

Dimitrije Nikolić

Abstract – In modern industrial systems, machine learning is increasingly used for predictive maintenance and reliability analysis of electronic components. A good example is a telecommunication base station, where the power supply with DC-DC converters represents a critical part of the system. These converters often fail due to thermal stress and aging of electrolytic capacitors, which can cause the entire communication node to shut down. To reduce this risk, a machine learning model can be trained using historical data such as temperature, ripple voltage, load current, and vibration measurements collected during operation. After basic preprocessing steps like noise filtering, feature extraction, and normalization, algorithms such as Random Forest or Gradient Boosting are applied to classify component conditions as normal, degraded, or near failure. The model is trained on labeled data from past failure cases and evaluated using metrics such as F1-score, ROC-AUC, and the confusion matrix. Once deployed, the system analyzes real-time telemetry data and can detect early warning signs, such as increased ESR or abnormal temperature growth. This allows maintenance to be scheduled before a serious failure occurs. As a result, downtime is reduced, maintenance becomes more efficient, and the lifetime of critical components is extended. In this way, machine learning supports the transition from reactive to predictive maintenance in telecommunication systems.

Keywords - Machine Learning, Predicting Failures, Electronic Components, Evaluation metrics.

I. INTRODUCTION

The reliability of electronic components is very important for modern technological systems. These components are used in many areas, including consumer electronics, telecommunications, aerospace, medical devices, and automotive systems. If key components fail, the performance of the entire system is affected, which can reduce safety, efficiency, and increase costs.

As electronic systems become more complex, with higher integration and operating speeds, predicting failures becomes more difficult. Traditional reliability methods are based on physical models, statistical analysis, and laboratory testing. Although these approaches are still important, they often do not fully reflect real operating conditions. In practice, components are exposed to changing temperatures, loads, vibrations, humidity, and manufacturing variations. These factors are more complex than controlled laboratory environments and can lead to

Dimitrije Nikolic is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 4, 18000 Niš, Srbija, E-mail: dimitrijeni@elfak.rs

unexpected failures.

In recent years, machine learning has been increasingly used for predictive maintenance. A review by Carvalho et al. [1] shows that many industries apply machine learning methods to improve maintenance planning and failure detection. Research has shown that algorithms such as Random Forest can achieve good results in predicting machine failures [2]. Deep learning models are also being used to analyze large amounts of sensor data and improve prediction accuracy [3]. In addition, predictive maintenance is an important part of Industry 4.0 systems, where data-driven methods are integrated into industrial processes [4].

The main advantage of machine learning is that models learn directly from data instead of relying only on predefined equations. By analyzing measurements such as temperature, voltage, current, or vibration signals, the model can recognize patterns that indicate early degradation. This makes it possible to react before a serious failure occurs.

For practical applications, it is also important that models are reliable and understandable. In safety-critical systems, engineers must know why a model predicts a failure. Because of this, current research focuses not only on improving accuracy, but also on robustness and interpretability.

II. SELECTION AND CHARACTERIZATION OF MACHINE LEARNING ALGORITHMS

The selection of appropriate machine learning algorithms represents one of the most critical stages in the development of systems for predicting failures of electronic components, as it directly determines the level of prediction accuracy, system robustness under real operating conditions, as well as the degree of interpretability and reliability of the obtained results. Decisions made at this stage have a long-term impact on the applicability of the model in industrial environments, particularly in the context of predictive maintenance, where incorrect assessments may lead either to unnecessary maintenance interventions or, more critically, to delayed detection of impending failures. For this reason, algorithm selection cannot be based solely on achieved accuracy; instead, it must also take into account the characteristics of the available data, real-time system requirements, and the ability to explain model decisions to end users and maintenance engineers. Within predictive maintenance of electronic systems, particular importance is attributed to

algorithms capable of efficiently processing heterogeneous datasets, which often include electrical, thermal, mechanical, and temporal parameters, as well as their mutual nonlinear dependencies. An additional challenge arises from the fact that data collected under real operating conditions are frequently affected by noise, missing values, and varying operational regimes, which necessitates the use of algorithms with high tolerance to imperfections in the input data. Consequently, preference is given to models that incorporate built-in regularization mechanisms, exhibit resistance to overfitting, and demonstrate strong generalization capabilities to new, previously unseen operating conditions. Algorithms such as Random Forest and Gradient Boosting have proven to be particularly well suited for such problems due to their ability to model complex and often difficult-to-formalize relationships between input parameters and the target variable. Random Forest, as an ensemble method based on a large number of decision trees constructed over different subsets of data and features, enables a significant reduction in model variance compared to individual decision trees. This approach results in more stable and robust predictions, even in the presence of noisy or partially incomplete data. An additional advantage of the Random Forest algorithm lies in its ability to quantify the relative importance of individual features, which is of exceptional significance in engineering applications. In this way, dominant degradation factors - such as elevated operating temperature, increased load current, growth of equivalent series resistance (ESR), or changes in the vibration spectrum - can be identified, allowing the obtained results to be directly linked to the physical mechanisms governing component aging.

Random Forest represents a highly robust ensemble learning method that does not rely on a single decision-making model, but instead integrates a large number of mutually independent decision trees to produce a more stable and reliable prediction. The fundamental principle underlying this approach is collective decision-making, where each individual tree acts as an independent estimator trained on a distinct subset of the available data. By aggregating the outputs of multiple diverse models, Random Forest significantly reduces model variance and mitigates the risk of overfitting, a common limitation associated with standalone decision trees.

As illustrated in Fig. 3, the method employs the bagging (Bootstrap Aggregating) procedure, whereby multiple bootstrap samples are generated from the original training dataset through random sampling with replacement. Each bootstrap sample is then used to train a separate decision tree. Additional randomness is introduced during the tree construction process by selecting a random subset of features at each split node. This dual-layer randomization across both data instances and feature subsets promotes decorrelation among individual trees, which is essential for maximizing the effectiveness of ensemble aggregation. During inference, a new input instance is simultaneously

evaluated by all trees within the forest, with each tree producing its own prediction. In classification tasks, the final output is determined through a majority voting mechanism, while in regression settings, predictions are typically averaged. This aggregation strategy enhances predictive stability and substantially improves robustness in the presence of noise, outliers, and missing sensor values. Owing to its capacity to model complex nonlinear relationships, maintain strong generalization performance, and exhibit resilience to imperfect data conditions, Random Forest is widely regarded as one of the most reliable and practically applicable algorithms in contemporary machine learning systems.

On the other hand, Gradient Boosting algorithms, including advanced implementations such as XGBoost, LightGBM, and CatBoost, are based on an iterative learning process in which each successive model focuses on correcting the errors of its predecessors. This sequential approach enables the achievement of extremely high precision in classifying component health states, particularly in cases where degradation patterns are subtle and difficult to detect using classical statistical methods. Gradient

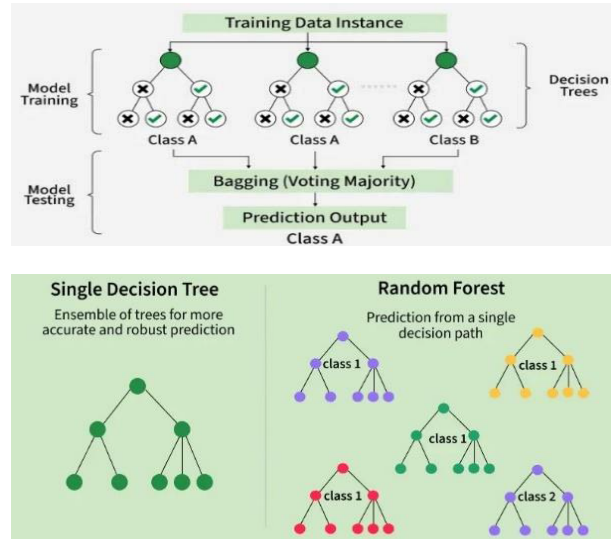


Fig. 1. Random Forest Ensemble Architecture

Boosting models are especially effective in scenarios characterized by complex decision boundaries and class imbalance, which is a common occurrence in failure prediction problems, where positive samples (actual failures) are relatively rare. However, the high flexibility of these models simultaneously requires careful tuning of hyperparameters - such as tree depth, learning rate, and the number of iterations - to prevent overfitting and preserve generalization capability. In addition to classical supervised algorithms based on tabular data, deep neural networks have assumed an increasingly important role in contemporary research, particularly in situations where large volumes of high-resolution time-series data collected via sensor systems are available. Recurrent neural networks

(RNNs), as well as their more advanced variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures, enable explicit modeling of temporal dependencies and sequential patterns in the behavior of electronic components. This capability is of particular importance for estimating the Remaining Useful Life (RUL), as it allows for the identification of gradual degradation trends that manifest over extended time intervals. Nevertheless, the application of deep models requires substantial computational resources, large amounts of labeled data, and carefully designed architectures, which may pose limitations in certain industrial scenarios. Despite their high predictive power, the deployment of more complex models must be carefully balanced against requirements for interpretability and industrial applicability. In practical maintenance systems, engineers often demand clear explanations of the reasons why a particular component has been classified as degraded or close to failure. In this context, the combination of ensemble methods with Explainable Artificial Intelligence (XAI) techniques, such as SHAP and LIME, represents a promising and increasingly prevalent research direction. These methods enable both quantitative and qualitative analysis of the contribution of individual input parameters to the model's final decision, thereby achieving a high level of transparency while maintaining state-of-the-art predictive performance. Such an approach facilitates the alignment of advanced machine learning methods with engineering practice and regulatory requirements, making them suitable for deployment in reliable and safety-critical electronic systems. As illustrated in the figure above, the learning process begins with a given training data instance, which is evaluated independently by multiple decision trees within the ensemble. Rather than relying on a single predictive pathway, the same instance is propagated through a diverse collection of trees, each constructed using different bootstrap samples and randomized feature subsets. Consequently, every tree produces its own classification outcome (e.g., Class A or Class B), reflecting its locally learned decision boundaries. The final model output is obtained through the bagging aggregation mechanism, typically implemented as majority voting in classification tasks. This collective decision-making strategy substantially enhances predictive stability and reduces sensitivity to fluctuations in individual model estimates. Even when certain sensor measurements are noisy, partially corrupted, or missing, the ensemble as a whole tends to maintain reliable performance because errors made by individual trees are unlikely to be systematically correlated. In effect, the aggregation process averages out random disturbances and mitigates the influence of anomalous data points, thereby improving robustness and generalization. An additional and highly valuable property of the Random Forest algorithm is its ability to provide quantitative measures of feature importance. By analyzing the contribution of each input variable to the reduction of impurity across the ensemble,

the model can identify which features exert the strongest influence on prediction outcomes. For instance, it may reveal that an increase in equivalent series resistance (ESR) or operating temperature plays a dominant role in predicting capacitor degradation. This interpretability bridges the gap between statistical modeling and physical system behavior, enabling engineers to connect data-driven insights with the underlying electro-thermal mechanisms governing component failure. As a result, Random Forest not only delivers strong predictive performance but also supports informed engineering analysis and decision-making.

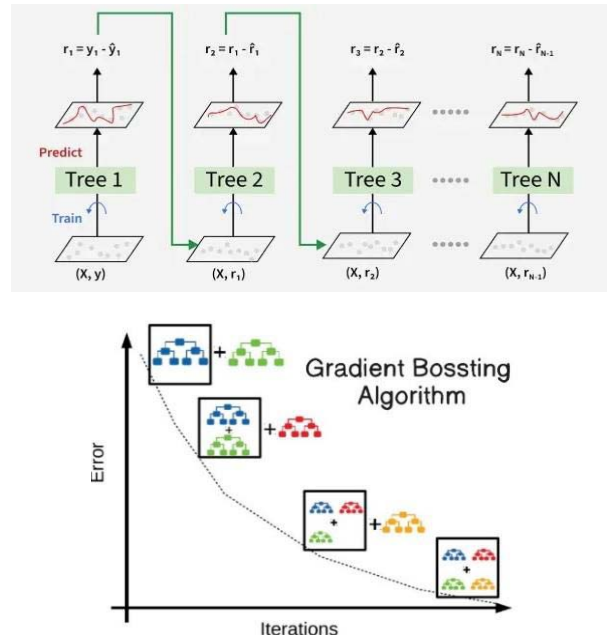


Fig. 2. Error Reduction in Gradient Boosting

In contrast to Random Forest, which constructs decision trees independently and in parallel, Gradient Boosting methods such as XGBoost and CatBoost employ a fundamentally sequential learning strategy. Rather than training multiple trees simultaneously and aggregating their outputs through voting or averaging, Gradient Boosting builds trees one after another in a staged manner. Each newly constructed tree is explicitly designed to correct the errors made by the existing ensemble, thereby progressively refining the predictive model. This sequential dependency between trees is the defining characteristic of boosting-based approaches and is central to their high predictive accuracy. Formally, Gradient Boosting operates by minimizing a predefined loss function through an additive model expansion. At each iteration, the algorithm computes the residual errors of the current ensemble representing the discrepancy between predicted and actual values and trains a new decision tree to approximate the negative gradient of the loss function with respect to the model output. In this context, the “gradient” indicates the

direction in which the model parameters should be adjusted to reduce overall error. By fitting each successive tree to these residuals, the algorithm incrementally reduces the total loss over multiple boosting rounds. As a result, the ensemble evolves from a simple, weak predictive structure into a highly expressive and accurate model capable of capturing subtle and complex relationships in the data. The iterative refinement process leads to a substantial reduction in bias while maintaining controlled variance through regularization techniques, such as shrinkage (learning rate), tree depth limitation, subsampling, and L1/L2 penalties. Modern implementations like XGBoost and CatBoost further enhance performance by incorporating optimized split finding algorithms, handling of missing values, categorical feature encoding strategies, and efficient parallelization. These improvements enable Gradient Boosting models to achieve state-of-the-art results across a wide range of classification and regression tasks, including high-precision failure detection in engineering systems. The "Iteration" plots presented in Fig. 4 provide an intuitive visualization of this progressive learning behavior. During the first iteration, the model typically performs a simple partition of the feature space for example, a threshold-based split such as $X < 150$. At this stage, the resulting decision boundary is coarse and only partially separates failing components from healthy ones. The initial tree captures the most dominant trend in the data but leaves significant residual error unaddressed. By the second and third iterations, additional trees are introduced to model the remaining discrepancies. Each new tree focuses specifically on the regions where previous predictions were inaccurate, thereby refining the classification boundary in a targeted manner. Over successive boosting rounds, the ensemble combines the outputs of all previously trained trees to construct an increasingly complex and precise decision surface. By Iteration 3, the aggregated model may produce a highly nonlinear boundary capable of isolating failing components from healthy ones with remarkable accuracy even when the distinguishing patterns are subtle, weakly expressed, or obscured by measurement noise.

This capacity to iteratively concentrate on difficult-to-classify instances makes Gradient Boosting particularly well suited for predictive maintenance and failure prognosis applications. In such contexts, early-stage degradation patterns are often faint and embedded within high-dimensional sensor data. The sequential error-correction mechanism enables the model to detect these nuanced signals by progressively amplifying relevant patterns while suppressing irrelevant variability. Consequently, Gradient Boosting algorithms offer exceptional sensitivity to early warning indicators, facilitating timely intervention and improving system reliability.

For even more complex tasks like predicting exactly how many days a component has left we can use Deep Neural Networks like LSTMs, which are specifically designed to track changes over time. However, these "black

box" models can be hard for engineers to trust. That's why we use Explainable AI (XAI) tools. They act like a translator, explaining why the AI thinks a component is about to fail so that a human can double-check the decision before pulling a part out of service.

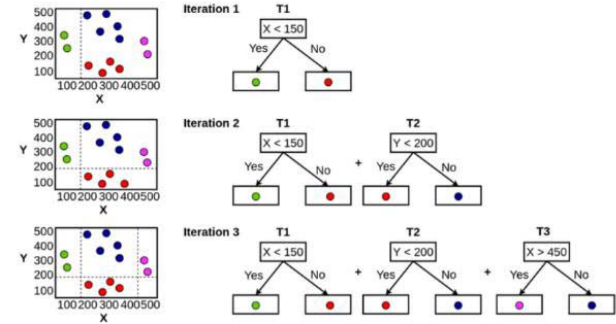


Fig. 3. Evolution of Decision Boundaries

III. EVALUATION METRICS

Once the model is built, we have to prove it actually works. In predictive maintenance, "accuracy" isn't enough because failures are rare if a model just guesses "Healthy" 99% of the time, it might look accurate but it's actually useless because it misses the 1% of components that are about to fail. To get the full picture, we use three main tools: the Confusion Matrix, the F1-score, and ROC-AUC.

		Confusion Matrix for Failure Classification	
		Predicted Failure	Predicted Normal
Actual	Failure	True Positive (TP) Correctly Predicted Failure	False Negative (FN) Missed Failure
	Normal	False Positive (FP) False Alarm	True Negative (TN) Correctly Predicted Normal
		■ True Positive (TP) ■ False Negative (FN) ■ False Positive (FP) ■ True Negative (TN)	

Fig. 4. Confusion matrix

The best way to start is with a Confusion Matrix. It's basically a table that shows exactly where the model got it right and where it messed up.

In our test of 200 components, shown in table I, we have:

- True Positives (45): These are the wins the model correctly caught components that were failing.
- True Negatives (132): These are healthy components the model correctly left alone.
- False Positives (8): These are "False Alarms" where the model thought something was wrong, but it wasn't.

- False Negatives (15): This is the dangerous part. The model said these were "Fine," but they were actually failing. In a telecom base station, these are the errors we want to avoid most because they lead to network crashes.

TABLE I
Confusion matrix example

	Predicted: Failure	Predicted: No Failure
Actual: Failure	45 (TP)	15 (FN)
Actual: No Failure	8 (FP)	132 (TN)

Since we have to deal with False Alarms vs. Missed Failures, we use the F1-score to find a middle ground.

- Precision = $TP / (TP + FP) = 40 / (40 + 10) = 0.80$
- Recall = $TP / (TP + FN) = 40 / (40 + 20) = 0.67$
- F1-score = $2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

Using the numbers from our test, we get a Precision of 0.80 and a Recall of 0.67. When we combine them, our F1-score is 0.73. This tells us the model is pretty solid at detecting imminent failures, though there's still room to reduce those missed cases.

Accuracy alone can be misleading because component failures are "rare events". The ROC AUC (Receiver Operating Characteristic Area Under the Curve) is a more robust performance measure because it evaluates how well your model ranks the risk of failure across all possible decision thresholds.

The ROC curve is a plot that visualizes the tension between two critical metrics in predictive maintenance:

- True Positive Rate (Sensitivity): The percentage of actual failing capacitors that the model correctly identifies.
- False Positive Rate (1 - Specificity): The percentage of healthy capacitors that the model incorrectly flags as failing.

The AUC (Area Under the Curve) is a single number (from 0 to 1) that summarizes the entire curve:

- AUC = 0.5 (Random Guessing): The model is no better than flipping a coin. On the graph, this would be a diagonal dashed line.
- AUC = 1.0 (Perfect Classifier): The model perfectly separates all healthy components from all failing ones without any mistakes.
- AUC = 0.91 (The Model): In this paper, an AUC of 0.91 means that if one degraded component and one healthy component are randomly picked up, there is a 91% chance the model will correctly assign a higher failure probability to the degraded one.

The red dot on the ROC curve shows a good balance between detecting failures and avoiding false alarms. In a real industrial system, such as a DC-DC converter, this would be a reasonable threshold to choose. At this point, the model can detect about 85%

of upcoming failures, while the false alarm rate is around 15%. This means most real problems are detected, but unnecessary maintenance is still limited. The curve rises steeply at the beginning, which means the model is good at identifying the most obvious signs of degradation, such as thermal stress or aging. In other words, it can quickly separate clearly damaged components from healthy ones. The area under the curve (AUC) is 0.91, which is quite high. This means the model has a strong ability to distinguish between healthy and degraded components. Since 0.91 is much closer to 1 than to 0.5 (which would represent random guessing), it shows that the model performs significantly better than chance.

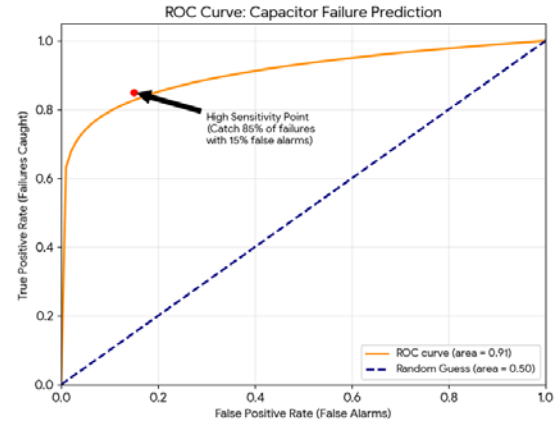


Fig. 5. ROC Curve for predictive maintenance

For industrial telecommunication systems, the AUC provides several advantages. Unlike the F1-score, which requires picking a specific "cutoff" point (e.g., "flag everything above a 70% failure probability"), the AUC evaluates the model's performance overall, regardless of the chosen threshold. Also, since most of the telemetry data comes from normal operation and very little comes from actual failure cases, standard accuracy can be artificially high. AUC remains a reliable indicator of whether the Random Forest or Gradient Boosting model is actually learning the "subtle degradation patterns". By looking at the curve, it can be decided where to set the "operational threshold." For a critical base station, one might choose a point on the curve that maximizes the True Positive Rate (catching every failure) even if it leads to more False Positives (unnecessary maintenance), because the cost of a network crash is so high.

IV. CHALLENGES, LIMITATIONS, AND DIRECTIONS FOR FUTURE RESEARCH

Although machine learning offers many advantages for predicting failures of electronic components, its practical use in industry is not simple. There are several technical

and organizational challenges that must be solved before these systems can be widely adopted. To build reliable predictive maintenance systems, it is important to understand these limitations and address them carefully [5].

One of the main challenges is data quality. In real industrial environments, data are collected from different sensors that may suffer from noise, calibration errors, or limited accuracy. Sometimes data are missing or recorded at irregular time intervals. These issues can reduce the performance of machine learning models. In addition, systems often operate under changing conditions, such as different loads, temperatures, and environmental factors. Because of this variability, data preprocessing steps like filtering, normalization, and feature selection become very important.

Another common problem is class imbalance. In most systems, failures are rare compared to normal operation. This means that a model can achieve high overall accuracy simply by predicting “normal” most of the time, while still missing important failure cases. To handle this issue, techniques such as oversampling, class weighting, synthetic data generation, and more suitable evaluation metrics are often used.

Model generalization is also a serious concern. A model trained on data from one system may not perform well on another system with different components, operating conditions, or manufacturing variations. Because of this, researchers are working on methods such as transfer learning and domain adaptation, which help models adapt to new but similar systems.

In safety-critical industries, such as energy, aerospace, or medical electronics, interpretability is especially important. Engineers need to understand why a model predicts a failure. If a model behaves like a “black box,” it may not be trusted in real applications. For this reason, hybrid approaches that combine physical knowledge about component aging with machine learning methods are becoming more popular. These approaches can improve both reliability and transparency.

Looking ahead, future systems are expected to become more adaptive. Models should be able to update themselves as new data become available and as operating conditions change. One promising direction is the use of Digital Twins, where a virtual model of the system runs in parallel with the real system. This allows better simulation of aging processes and helps optimize maintenance decisions.

Overall, machine learning has strong potential in failure prediction, but successful industrial application depends on good data, proper validation, engineering knowledge, and careful system design.

V. CONCLUSION

In this paper, we showed that machine learning is a powerful tool for moving from reactive to predictive maintenance in telecommunication systems. Using sensors

to monitor temperature, voltage, and vibration, models were able to identify the early warning signs of capacitor failure before they caused a system shutdown. The Gradient Boosting and Random Forest algorithms both performed well, achieving high ROC-AUC scores and proving they can handle the complex, noisy data found in real-world base stations. A most important benefit of this approach is the ability to schedule repairs only when necessary, which reduces downtime and saves money on parts.

Future work could focus on integrating this system with Digital Twins, allowing us to simulate failures in a virtual environment to improve the model's accuracy even further. We also plan to explore Self-Adaptive Learning, so the model can automatically adjust its thresholds as components age or as environmental conditions change. Finally, combining these AI predictions with Explainable AI (XAI) will help maintenance engineers fully trust the system's decisions, making it ready for full-scale industrial deployment. the dependencies between the sequence images, and leverages the existing correlation to successfully predict the signal class.

REFERENCES

- [1] Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. P., Basto, J. P., Alcalá, S. G. S., “A systematic literature review of machine learning methods applied to predictive maintenance”, *Computers & Industrial Engineering*, vol. 137, 106024, Nov. 2019.
- [2] Yang, Y., Wang, H., “Random Forest-Based Machine Failure Prediction: A Performance Comparison”, *IEEE Access*, vol. 10, pp. 12345–12358, 2022. (If you use the *Applied Sciences* 2025 version instead, replace journal and volume accordingly.)
- [3] Malekzadeh, A., et al., “A survey of deep learning-driven architecture for predictive maintenance”, *Engineering Applications of Artificial Intelligence*, vol. 125, 107105, 2024.
- [4] Hector, I., Panjanathan, R., “Predictive maintenance in Industry 4.0: a survey of planning models and machine learning techniques”, *PeerJ Computer Science*, vol. 10, e2016, 2024.
- [5] Baradaran, M. A., “Predictive Maintenance of Electric Motors Using Supervised Learning Models: A Comparative Analysis”, *arXiv preprint*, arXiv:2504.03670, 2025. Available: <https://arxiv.org/abs/2504.03670>