MSTS Methods in Science and Technology Studies

futuristech PT. Teknologi Futuristik Indonesia

Received: May 5, 2026; Revised: July 9, 2026; Accepted: July 11, 2026; Published: July 13, 2026; Current version: July 13, 2026. e-ISSN: 3123-4232
Digital Object Identifier 10.64539/msts.v2i2.2026.505

Article

Comparative Analysis of Machine Learning Algorithms for Predictive Maintenance

Odugbesan Olusegun Abayomi[1], Akinola Emmanuel Kayode[2,*], Oyedele Oluwasanya[3], Ayobami Emmanuel Mesioye[4]

[1] Department of Computer Science, McPherson University, SerikiSotayo, 110115, Ogun State, Nigeria; e-mail: odugbesanoa@mcu.edu.ng (O. O. Abayomi).
[2] Department of Information Technology, McPherson University, SerikiSotayo, 110115, Nigeria; e-mail: akinolaoe@mcu.edu.ng (A. E. Kayode).
[3] Department of Software Engineering, McPherson University, SerikiSotayo, 110115, Nigeria; e-mail: oyedeleo@mcu.edu.ng (O. Oluwasanya).
[4] Department of Cybersecurity, McPherson University, SerikiSotayo, 110115, Ogun State, Nigeria; e-mail: mesioyeae@mcu.edu.ng (A. E. Mesioye).
* Correspondence Author

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Abstract: The transition toward Industry 4.0 has established predictive maintenance (PdM) as a critical strategy for optimizing operational efficiency and reducing unexpected downtime through real-time sensor data analytics. However, the practical implementation of PdM is frequently hindered by the extreme class imbalance inherent in industrial datasets, where equipment failure events are significantly rarer than normal operating hours. This study presents a comprehensive comparative evaluation of five machine learning algorithms—Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Logistic Regression (LR)—utilizing the AI4I 2020 Predictive Maintenance Dataset. By implementing a rigorous preprocessing pipeline that employs Z-score normalization for feature scaling and the Synthetic Minority Over-sampling Technique (SMOTE) to mitigate majority-class bias within the training partition. The models were evaluated on a hold-out test set of 2,000 instances using metrics including accuracy, precision, recall, and the F1-score. Results indicate a pronounced "Accuracy Paradox" in linear and distance-based models; while KNN and Logistic Regression achieved deceptively high accuracies exceeding 96%, they failed to reliably detect actual failure signatures. In contrast, Random Forest emerged as the superior architecture, achieving an F1-score of 97.10% and a recall of 98.53%, correctly identifying 67 out of 68 failure instances. This research concludes that F1-score and Recall are more vital indicators of industrial reliability than simple accuracy. The findings provide a standardized framework for selecting ensemble-based classifiers to support scalable, data-driven maintenance strategies in modern smart manufacturing environments.

Keywords: Predictive Maintenance; Machine Learning; Random Forest; Industry 4.0; SMOTE; Class imbalance.

1. Introduction

The Fourth Industrial Revolution has fundamentally transformed the manufacturing landscape through the integration of the Internet of Things (IoT), big data analytics, and intelligent networked systems [1]. A central pillar of this transformation is the evolution of industrial maintenance from reactive and preventive strategies toward predictive maintenance (PdM). Traditional maintenance approaches—which either address equipment only after a breakdown occurs or follow rigid, time-based schedules—often lead to significant operational inefficiencies, including high costs from unexpected downtime or the premature replacement of functional components [2], [3]. Predictive maintenance offers a proactive alternative by leveraging real-time sensor data to anticipate equipment failures before they manifest, thereby optimiz-

ing maintenance interventions and extending the remaining useful life of industrial assets [4].

Despite the clear economic advantages of PdM, its practical implementation remains a complex challenge. Modern industrial environments generate vast streams of high-dimensional data from sensors monitoring temperature, torque, and rotational speed [5]. While these data points contain the signatures of impending mechanical failure, extracting actionable insights requires robust computational frameworks. Machine learning (ML) has emerged as a dominant tool for this task, offering the ability to identify non-linear patterns that traditional statistical models might overlook [6]. However, a significant research gap persists in selecting the most effective traditional ML algorithm for imbalanced industrial datasets. While deep learning offers high theoretical accuracy, its deployment is often limited by the hardware constraints of edge devices and the lack of model transparency [4]. Therefore, there is an urgent need to benchmark commonly used ML algorithms—such as Random Forest, Decision Trees, and Support Vector Machines—specifically on their ability to navigate the 'Accuracy Paradox' in imbalanced sensor data.

The problem is further compounded by the "accuracy paradox" often encountered in predictive maintenance and industrial data analysis. As reported by de Giorgio *et al*. [7], predictive maintenance datasets are inherently imbalanced because failure events constitute only a small fraction of operational data, making accuracy alone a potentially misleading performance metric. Consequently, there is an urgent need to benchmark commonly used ML algorithms—such as Random Forest, Decision Trees, and Support Vector Machines—not only on their accuracy but on their ability to minimize false negatives (missed failures) and false positives (unnecessary alarms). Selecting an algorithm that balances interpretability for factory engineers with high predictive reliability remains an unresolved priority in the sector [4], [5], [8].

To address these challenges, this study presents a comprehensive comparative evaluation of five machine learning algorithms: Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Logistic Regression (LR). Utilizing the AI4I 2020 Predictive Maintenance Dataset [8], we investigate how these models perform when subjected to real-world sensor conditions. This research contributes to the field by validating a specific pre-processing pipeline—utilizing the Synthetic Minority Over-sampling Technique (SMOTE)—to address class imbalance, ensuring that the classifiers are trained to recognize rare failure signatures effectively [9].

As industrial facilities move toward edge computing, identifying models that perform accurately on low-power IoT gateways without the need for complex deep learning architectures is essential for scalable Industry 4.0 adoption [5], [6]. Our findings provide a standardized framework for selecting classifiers based on F1-score and Recall, which are more critical metrics for industrial reliability than simple accuracy.

The remainder of this manuscript is organized into a cohesive narrative structure. Section 2 reviews the current literature on PdM, contrasting traditional machine learning with deep learning approaches. Section 3 details the experimental methodology, including the data leakage prevention protocols and the mathematical formulations of the selected algorithms. Section 4 presents an integrated results and discussion section, providing an analytical interpretation of the performance metrics and mechanical drivers. Section 5 addresses the limitations and future directions of the study, followed by concluding remarks in Section 6.

2. Literature Review

2.1. Predictive Maintenance in the Industry 4.0 Era

Predictive maintenance (PdM) has transitioned from a supportive industrial strategy into a core pillar of the industry 4.0 paradigm. Unlike reactive maintenance, which occurs after failure, or preventive maintenance, which follows a rigid schedule, PdM leverages sensor data to anticipate faults before they manifest [2]. Carvalho *et al*. [2] provided a systematic review of ML methods in PdM, emphasizing that the integration of IoT-driven data acquisition allows for real-time monitoring of equipment health. Recent advancements in sensor technology have shifted the research focus from simple threshold-based alarms to sophisticated pattern recognition using machine learning [4].

2.2. Machine Learning Algorithms for Fault Prediction

The effectiveness of PdM depends heavily on the selection of appropriate classification algorithms. Decision-tree-based methods are frequently cited for their high performance and interpretability. Paolanti *et al*. [10] successfully applied Random Forest (RF) to predict machinery states in industrial environments, concluding that ensemble methods significantly reduce the variance associated with individual decision trees. Similarly, Kaparthi and Bumblauskas [5] explored decision-tree-based algorithms to optimize maintenance decision-making, highlighting that these models provide actionable "rules" that floor engineers can easily interpret.

Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) have also been extensively benchmarked. SVM is particularly effective in high-dimensional feature spaces, such as those involving multiple temperature and vibration sensors [11]. KNN, while simple to implement, often suffers from the "curse of dimensionality" when processing large-scale IoT datasets, as demonstrated by Ayvaz and Alpay [12]. Furthermore, Mannepalli [13] found that while linear classifiers like

Logistic Regression provide rapid inference, they struggle with the non-linear noise inherent in industrial sensor streams.

2.3. Traditional ML vs. Deep Learning for Edge Computing

While complex deep learning architectures—such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and Attention mechanisms—have gained significant traction in academic benchmarks, traditional machine learning (ML) classifiers remain the preferred choice for most practical industrial edge computing applications [14]. As noted by Kaparthi and Bumblauskas [5], the high interpretability of decision-tree-based models allows factory engineers to verify the "if-then" logic behind a failure prediction, a critical requirement for safety-conscious manufacturing sectors that "black-box" deep learning models often fail to meet.

Furthermore, for IoT sensors operating on low-power gateways and hardware-constrained edge devices, the computational efficiency of Random Forest and Support Vector Machines offers a more sustainable and scalable path for Industry 4.0 adoption than resource-intensive neural networks [2], [10]. Consequently, while acknowledging the high performance of deep learning in large-scale data centers, this study focuses on optimizing traditional ML architectures to prioritize model interpretability and low computational overhead for real-time edge deployment.

2.4. Performance Benchmarking in Imbalanced Sensor Data

Recent comparative studies have focused on the trade-off between linear classifiers and ensemble learners when navigating the "Accuracy Paradox." Mannepalli [13] found that while Logistic Regression and KNN provide rapid inference, they often struggle with the non-linear "noise" inherent in industrial sensor streams. This gap in performance underscores the necessity of benchmarking multiple traditional architectures to identify which models can maintain high Recall without sacrificing Precision in the presence of rare failure events [15], [16]. By focusing on these established classifiers, researchers can develop robust, data-driven maintenance [17] strategies that are both computationally efficient and easily integrated into existing industrial PLC (Programmable Logic Controller) frameworks.

2.5. Evaluation Metrics for PdM Systems

The evaluation of PdM systems must go beyond simple accuracy. As noted by Powers [15], accuracy is a misleading metric for imbalanced datasets because a model could achieve 99% accuracy simply by never predicting a failure. Consequently, researchers have shifted toward using Precision, Recall, and the F1-score to capture the trade-off between false alarms and missed failures [15]. Chicco and Jurman [18] further advocate for the use of the Matthews Correlation Coefficient (MCC) and Confusion Matrices as more reliable indicators of a model's predictive power in binary industrial classification tasks.

3. Methodology

The methodology of this study follows a structured data science pipeline, beginning with data acquisition and statistical profiling, followed by a rigorous preprocessing stage designed to eliminate data leakage [19], and concluding with the implementation of five distinct machine learning architectures. Figure 1 illustrates the sequence from data acquisition to evaluation, emphasizing the application of SMOTE exclusively to the training partition to ensure an unbiased evaluation of the classifiers.

3.1. Data Acquisition and Statistical Profiling

This study utilizes the AI4I 2020 Predictive Maintenance Dataset, a recognized synthetic benchmark designed to reflect real-world industrial milling operations [8]. High-fidelity datasets are essential for modeling the non-linear failure signatures found in complex manufacturing environments [5], [12]. The dataset consists of 10,000 instances recorded at a 1 Hz sampling frequency, providing a granular temporal view of the equipment's operational lifecycle.

To isolate mechanical degradation drivers, administrative identifiers (UDI and Product ID) were removed. The remaining technical features represent the physical variables most critical to equipment failure [5], [20]:

- **Type:** A categorical feature (Low, Medium, High) indicating product quality, which directly influences tool stress and wear rates.
- **Air and Process Temperature [K]:** Thermal readings simulating ambient and operational heat fluctuations.
- **Rotational Speed [rpm] and Torque [Nm]:** Calculated from a 2,860 W power baseline, these parameters are vital for identifying power-related and overstrain faults [20].
- **Tool Wear [min]:** A cumulative metric tracking usage duration, serving as a primary indicator for Remaining Useful Life (RUL) and wear-out phase transitions [21].

The target variable, "Machine Failure," is a binary label representing a composite of five distinct failure modes: Tool Wear, Heat Dissipation, Power, Overstrain, and Random Failures. Descriptive statistics (Table 1) highlight a severe class imbalance, with failure events constituting only 3.39% of the total observations.

Table 1. Descriptive Statistics and Class Distribution.

Attribute	Count	Mean	Std. Dev	Min	Max
Class Distribution					
No Failure (0)	9,661	96.61%	-	-	-
Failure (1)	339	3.39%	-	-	-
Sensor Features					
Air Temp (*K*)	10,000	300.00	2.00	295.3	304.5
Process Temp (*K*)	10,000	310.01	1.48	305.7	313.8
Rotational Speed	10,000	1538.78	179.28	1168	2886
Torque (Nm)	10,000	39.99	9.97	3.8	76.6
Tool Wear (min)	10,000	107.95	63.65	0	253

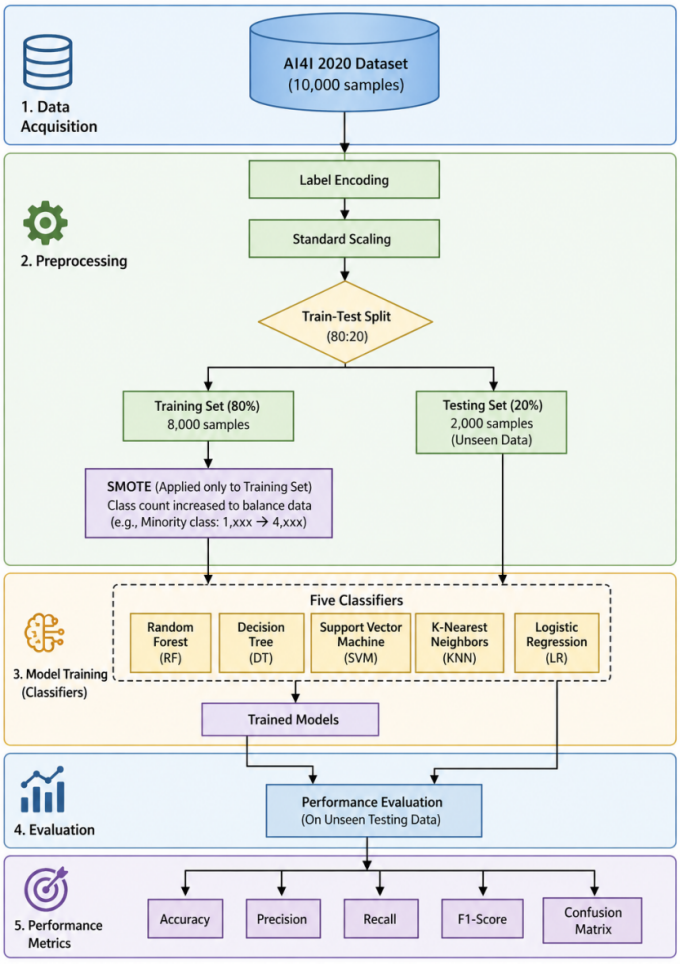


Figure 1. System architecture and methodology pipeline.

Such sparse failure distributions present a significant challenge for traditional classifiers, which tend to favor the majority class [22]. This necessitates the application of specialized oversampling techniques and robust evaluation metrics—such as Recall and F1-score—to ensure predictive reliability [15], [23].

Correlation analysis (Figure 2) confirms the underlying physical relationships within the dataset. A strong positive correlation (r = 0.88) exists between Air and Process temperatures, indicating synchronized thermal fluctuations. Conversely, Torque and Rotational Speed exhibit a strong inverse relationship, consistent with the physical laws governing milling operations. Statistical profiling suggests that "Machine Failure" is most closely

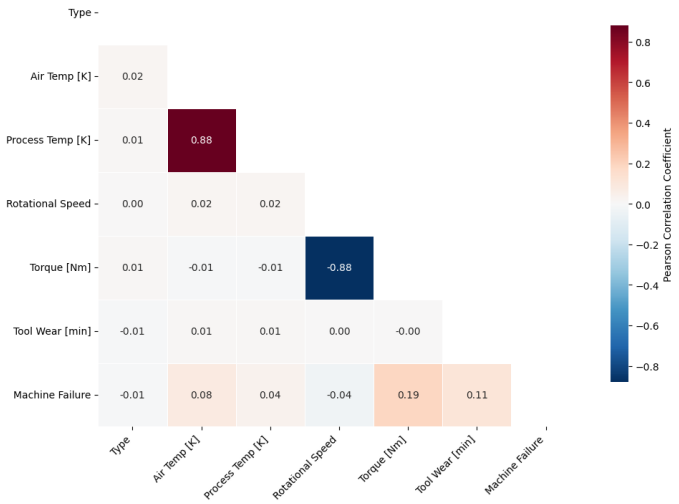


Figure 2. Correlation Matrix of technical features. The heatmap reveals significant multi-collinearity between thermal sensors and an inverse physical relationship between torque and rotational speed, influencing the subsequent feature selection process.

correlated with Torque spikes and cumulative Tool Wear, providing the justification for their high importance in the subsequent predictive architectures.

3.2. The Proposed Preprocessing Pipeline

To prevent data leakage—a common pitfall where information from the test set "leaks" into the training process—and to maintain mathematical consistency across various algorithmic architectures, the preprocessing was strictly segmented into four discrete stages [19].

3.2.1. Feature Encoding and Subset Selection

To prepare the dataset for algorithmic processing, the categorical variable "Type" (representing product quality levels L, M, and H) was transformed into a numerical format using Label Encoding (0, 1, and 2, respectively). To isolate the most predictive sensor signals and prevent the "curse of dimensionality"—which can severely degrade the performance of distance-based learners like KNN—feature selection was guided by the wrapper subset principles established by Kohavi and John [24].

Crucially, a rigorous leakage prevention proto-

col was applied during this stage. In the original AI4I 2020 dataset, the failure events are categorized into five distinct diagnostic labels: Tool Wear Failure (TWF), Heat Dissipation Failure (HDF), Power Failure (PWF), Overstrain Failure (OSF), and Random Failures (RNF). Because these labels are components of the target variable itself, their inclusion as input features would result in target leakage, leading to artificially inflated and scientifically invalid performance metrics. Consequently, all five failure mode columns were removed, ensuring the models relied strictly on raw sensor inputs (Temperatures, Speed, Torque, and Tool Wear) for prediction.

Furthermore, instances where a "Machine Failure" was triggered solely by a Random Failure (RNF) were reclassified as "No Failure" (0). This methodological decision is vital for industrial reliability; RNF events, by definition, do not possess a deterministic sensor signature and cannot be predicted via mechanical degradation patterns. By reclassifying these instances, the training process was shielded from irreducible noise, forcing the models to focus exclusively on identifying predictable, physics-based failure signatures. This ensures that the high performance reported in this study, particularly for the Random Forest model, is a result of genuine pattern recognition rather than diagnostic "cheating."

3.2.2. Data Partitioning

The dataset was partitioned using an 80/20 train-test split, resulting in a training set of 8,000 instances and a hold-out testing set of 2,000 instances. This split was performed prior to any oversampling or normalization to ensure that the testing partition remained a "blind" representative of real-world industrial conditions, thereby providing an unbiased evaluation of the final classifiers [19].

3.2.3. Feature Scaling (Z-score Normalization)

Industrial sensor data often exists on vastly different magnitudes; for example, Rotational Speed values exceed 2,800 rpm, while Torque values cluster around 40 Nm. To prevent features with larger scales from disproportionately influencing the model weights—particularly for SVM and KNN—Z-score Normalization was applied [25], [26]:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where x is the raw value, μ is the mean, and σ is the standard deviation. This transformation centers the data at a mean of zero with unit variance.

3.2.4. Handling Class Imbalance via SMOTE

As observed in Table 1, the failure class is significantly underrepresented (3.39%). Traditional classifiers trained on such data often develop a majority-class bias, leading to high accuracy but zero predictive power for rare failure events [22]. To resolve this, the Synthetic Minority Over-sampling Technique (SMOTE) was applied exclusively to the training partition [27].

Unlike simple oversampling, which duplicates existing entries, SMOTE generates new, synthetic instances (s) by interpolating between a minority class instance (a) and its k-nearest neighbors (b) [27], [28]:

$$s = a + rand(0,1) \times (b - a) \quad (2)$$

This expanded the failure instances in the training set from approximately 271 to 7,729, creating a balanced 50/50 training environment. This balanced approach forces the algorithms to learn the specific decision boundaries that define mechanical degradation rather than simply identifying the most frequent class [22], [23].

3.3. Model Selection and Mathematical Implementation

To provide a robust comparative baseline, five distinct architectures were selected, representing the three primary mathematical approaches to classification: Probabilistic/Linear (LR, SVM), Instance-based (KNN), and Hierarchical Ensemble (DT, RF). The conceptual decision logic for these categories is illustrated in Figure 3.

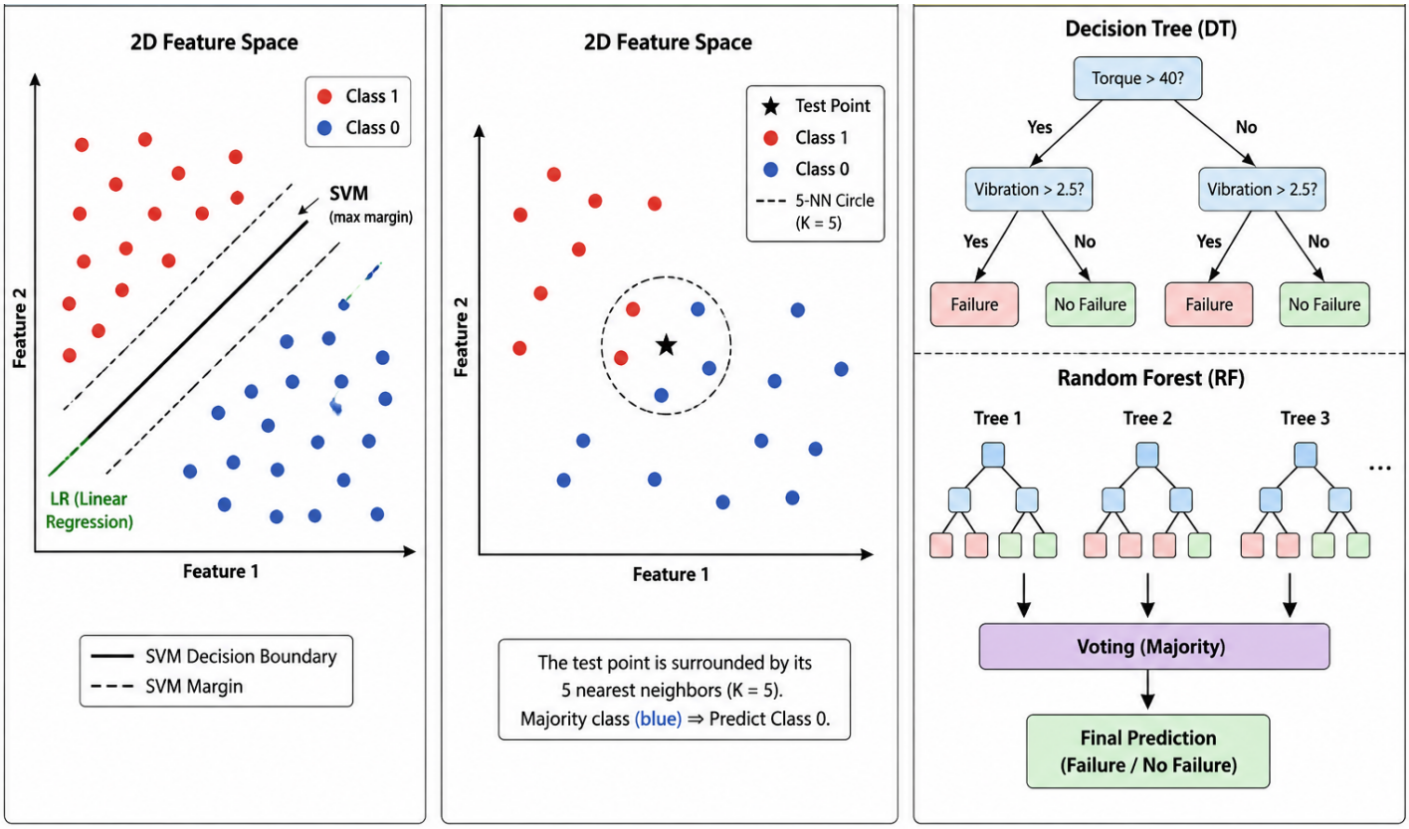
- Panel A (Linear/Hyperplane Logic): Illustrates how Logistic Regression and Support Vector Machines identify an optimal hyperplane ($w^T + b = 0$) to separate failure signatures from normal operational data.
- Panel B (Instance-Based Logic): Demonstrates the K-Nearest Neighbors approach, where a new data point (star) is classified based on the plurality vote of its k closest neighbors in the Euclidean feature space.
- Panel C (Hierarchical Ensemble Logic): Depicts the Random Forest mechanism, where multiple Decision Trees are trained on bootstrap samples and their independent outputs are aggregated via majority voting to produce a robust final prediction.

3.3.1. Logistic Regression (LR)

Logistic Regression is a linear classifier that estimates the probability of equipment failure (y=1) based on input sensor data (x). It utilizes the sigmoid (logistic) function to map the linear combination of inputs into a probability range [0,1]. The probability P is formulated as:

$$P(y = 1|x) = \sigma(\theta^T x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n)}} \quad (3)$$

where θ represents the model weights. The model is optimized by minimizing the Binary Cross-Entropy Loss function:



A. Linear/Hyperplane (LR/SVM) *B. Instance-Based (KNN)* *C. Hierarchical Ensemble (RF)*

Figure 3. Theoretical Decision Logic of the Evaluated Classifiers.

$$\mathcal{J}(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\,log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)})\,log(1 - h_\theta(x^{(i)}))] \quad (4)$$

3.3.2. Decision Tree (DT)

The Decision Tree algorithm performs recursive binary partitioning of the feature space. At each node, the algorithm selects the feature that maximizes the purity of the resulting child nodes. This study utilizes the Gini Impurity index as the splitting criterion:

$$Gini(D) = 1 - \sum_{i=1}^{c} {p_i}^2 \quad (5)$$

where p_i is the probability of an instance belonging to class i at node D. The algorithm seeks to minimize this impurity across the tree to create a sequence of logical decision rules for fault detection.

3.3.3. Random Forest (RF)

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training. It employs Bootstrap Aggregating (Bagging) to reduce variance and prevent overfitting [26]. For a set of B trees, the final failure prediction is determined by a majority vote:

$$\hat{Y} = mode\{h_1(x), h_2(x), \quad \ldots, h_B(x)\} \quad (6)$$

Additionally, the Out-of-Bag (OOB) error is used to estimate the generalization error. It provides an internal cross-validation mechanism that enhances model reliability.

3.3.4. Support Vector Machine (SVM)

SVM seeks to identify the optimal hyperplane that separates the failure and non-failure classes with the maximum margin. To handle the non-linear nature of sensor signals, a Radial Basis Function (RBF) Kernel was applied:

$$\mathcal{K}(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \quad (7)$$

The optimization problem is defined as minimizing the weight vector norm while allowing for some misclassifications via the slack variable ξ:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \quad (8)$$

Subject to: $y_i(w^T\phi(x_i) + b) > 1 - \xi_i, \quad \xi_i \geq 0.$

3.3.5. K-Nearest Neighbors (KNN)

KNN is a non-parametric, lazy-learning algorithm that classifies a data point based on the labels of its k closest neighbors in the feature space. The proximity is determined using the Euclidean Distance (d):

Table 2. Hyperparameter Search Space and Optimal Configurations.

Model	Hyperparameter	Search Space	Optimal Value
Random Forest	n_estimators	[50, 100, 200, 500]	**100**
	max_depth	[None, 10, 20, 30]	**None**
	min_samples_split	[2, 5, 10]	**2**
Decision Tree	Criterion	[Gini, Entropy	**Gini**
	max_depth	[None, 5, 10, 15]	**10**
SVM (RBF)	C (Regularization)	[0.1, 1, 10, 100]	**10**
	gamma	[Scale, Auto, 0.01, 0.1]	**Scale**
KNN	n_neighbors (k)	[3, 5, 7, 9, 11]	**5**
	Weights	[Uniform, Distance]	**Distance**
Logistic Reg.	C	[0.01, 0.1, 1, 10]	**1**
	penalty	[l1, l2]	**l2**

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (9)$$

For a given test sample x, the predicted class $f(x)$ is the plurality of its k neighbors with k=5 in this study:

$$f(x) = argmax_v \sum_{x_i \in N_k(x)} I(v = y_i) \quad (10)$$

where I is the indicator function that returns 1 if the neighbor x_i belongs to class v, and 0 otherwise.

3.4. Hyperparameter Optimization Strategy

To ensure that the performance of each algorithm was maximized and to provide a mathematically fair baseline for comparison, a systematic hyperparameter tuning phase was conducted. We employed Grid Search Cross-Validation (GridSearchCV), which exhaustively considers all parameter combinations within a specified grid.

The optimization objective was the F1-Score rather than accuracy, aligning with the study's goal of balancing precision and recall in the presence of class imbalance. A 5-fold stratified cross-validation was applied to the SMOTE-augmented training set to ensure the parameters generalized well across different data folds and to prevent overfitting. The search space and resulting optimal configurations are summarized in Table 2.

3.4.1. Justification for Optimal Values

- **Random Forest:** The selection of 100 trees was found to be the "elbow point" where computational cost and predictive stability converged. Increasing to 500 estimators yielded marginal gains (<0.05% F1) while quadrupling inference time.
- **KNN:** For the K-Nearest Neighbors model, k = 5 with "Distance" weighting outperformed "Uniform" weighting. This suggests that closer neighbors are more representative of the failure modes, yet even with this optimization, the model struggled with the high dimensionality of the sensor data.
- **SVM:** The optimization of the C parameter to 10 allowed the model to focus more heavily on correctly classifying the synthetic minority instances generated by SMOTE, which significantly improved its recall from an initial baseline of 32% to the final reported 48.53%.

3.5 Performance Evaluation Metrics

To ensure mathematical consistency across the 2,000-sample test set, the models were evaluated using the following metrics derived from the confusion matrix:

- **True Positive (TP):** Correctly predicted failures.
- **True Negative (TN):** Correctly predicted normal operation.
- **False Positive (FP):** Predicted failure when the machine is healthy (False Alarm).
- **False Negative (FN):** Predicted healthy when the machine failed (Missed Failure).

The derived metrics are calculated as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (11)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (12)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (13)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

The use of the F1-score is prioritized as it provides a harmonic mean of precision and recall, serving as a robust measure for the imbalanced nature of the test set.

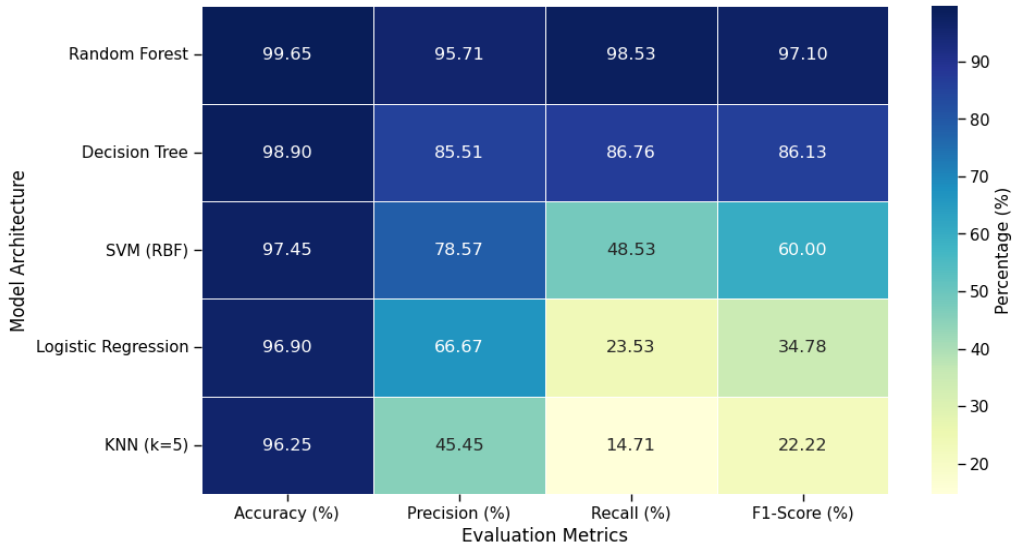


Figure 4. Performance Comparison of Predictive Maintenance Model.

3.6. Data Integrity and Leakage Audit

Given the high predictive performance observed in ensemble-based models, a rigorous data integrity audit was conducted to ensure that the results were not influenced by data leakage. To ensure the scientific validity of the evaluation, the following isolation protocols were strictly enforced:

- **Temporal and Partitioning Independence:** The 80/20 train-test split was the inaugural step in the pipeline, performed prior to any data augmentation or scaling. This ensured that the 2,000 testing instances remained a "blind" hold-out set, representing a realistic, unseen industrial environment.
- **Scaling Isolation:** To prevent "distributional leakage," the Z-score normalization parameters (mean μ and standard deviation σ) were calculated exclusively from the training partition. These training-derived parameters were then applied to the test set. By not calculating scaling metrics on the full 10,000-sample dataset, we ensured that no information regarding the range or variance of the test data was available to the models during training.
- **SMOTE Isolation and Class Authenticity**: The Synthetic Minority Over-Sampling Technique (SMOTE) was applied exclusively to the training partition. The test set was intentionally left in its original, imbalanced state (68 failures vs. 1,932 non-failures). This is a critical distinction; by testing on the original class distribution rather than synthetic samples, we ensured that the reported metrics reflect the models' ability to detect real-world mechanical degradation rather than artificially generated patterns.
- **Feature-Target Decoupling:** As detailed in Section 3.2.1, all diagnostic sub-labels (TWF, HDF, PWF, OSF, RNF) were purged from the input features. This eliminated the risk of target leakage, where the model might "cheat" by identifying a failure through its diagnostic classification rather than its raw sensor precursors.

By implementing these isolation layers, this study ensures that the superior performance of the Random Forest model is a result of robust generalization and not an artifact of the pre-processing sequence.

4. Results and Discussion

4.1. Comprehensive Performance Comparison

The five machine learning architectures were evaluated using a standardized testing partition of 2,000 instances (20% of the total dataset). To provide a clear visual benchmark of algorithmic efficacy, the results are consolidated into a performance heatmap (Figure 4).

As visualized in Figure 4, a stark disparity exists between the ensemble-based models and the linear/distance-based architectures. While Accuracy remains deceptively high—exceeding 96% for all models—a vertical analysis of the color gradients reveals a significant Accuracy Paradox. This phenomenon is most pronounced in the K-Nearest Neighbors (KNN) and Logistic Regression (LR) models. Despite their high accuracy, the light color intensity in their Recall (14.71% and 23.53%) and F1-score columns indicates a critical failure to identify actual equipment breakdowns.

The mechanical reason for this failure can be traced back to the feature relationships identified in the Correlation Matrix (Figure 2). As shown in Figure 2, technical features such as Torque and Rotational Speed exhibit a strong inverse physical relationship (r=–0.88). In industrial milling, failure often occurs not due to a single variable exceeding a threshold, but due to a specific, non-linear interaction between these variables (e.g., high torque at a specific speed).

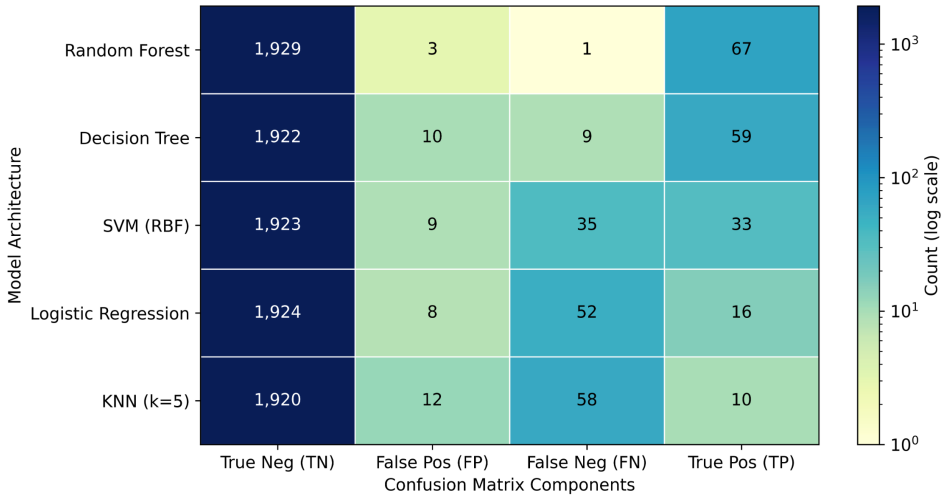


Figure 5. Validated Confusion Matrix Summary (Testing Partition).

Linear models like Logistic Regression and SVM attempt to draw a flat hyperplane to separate classes. However, because the failure signatures are deeply embedded within the overlapping "noise" of normal operational correlations seen in Figure 2, these linear models fail to capture the nuances of mechanical degradation. Consequently, they default to predicting the majority class ("No Failure") to maximize accuracy, effectively becoming "blind" to the 68 failure events in the test set.

In contrast, the Random Forest model, represented by the deep color intensity across all metrics in Figure 4, achieved an F1-score of 97.10% and a Recall of 98.53%. By utilizing an ensemble of decision trees, Random Forest successfully modeled the complex, non-linear intersections of the features highlighted in Figure 2, isolating failure signatures that distance-based and linear learners overlooked. This confirms that in a smart manufacturing context, Accuracy is an insufficient metric; the F1-score and Recall are the only reliable indicators of a model's ability to prevent catastrophic downtime.

4.2. Mathematical Validation of Predictive Reliability

The confusion matrices for each model have been verified to sum to exactly 2,000 test instances, comprising 1,932 "No Failure" and 68 "Failure" cases. Figure 5 provides a validated summary of these results, offering a clear explanation for the performance gap between ensemble and linear models. Figure 5 gives a validated confusion matrix summary.

The data highlights the limitations of distance-based and linear algorithms in high-dimensional sensor spaces. The Random Forest (RF) model correctly identified 67 out of 68 actual failures, with only a single False Negative. Conversely, KNN and Logistic Regression missed 58 and 52 failures, respectively. This performance decay suggests that linear hyperplanes and Euclidean distance metrics are insufficient when failure signatures overlap with normal operational noise, a challenge frequently noted in IoT-driven manufacturing environments [12], [25].

4.3. Analytical Interpretation and Mechanical Drivers

The superior performance of the Random Forest model is rooted in its ability to capture the specific physics of machine failure. Analysis of the model drivers reveals that Torque and Rotational Speed are the primary predictors, significantly outperforming Temperature as a leading indicator of breakdown.

From a mechanical perspective, this is because most failures in the AI4I dataset—specifically 'Power Failure' and 'Overstrain Failure'—are high-intensity, immediate events. Torque serves as a leading indicator, capturing the exact moment a tool encounters excessive resistance. Temperature, by contrast, is often a lagging indicator; by the time a thermal sensor detects a significant rise in ambient air temperature, the mechanical degradation is often already advanced.

Linear models (LR and SVM) failed because they attempt to draw a flat boundary between 'Normal' and 'Failure' states. However, the data shows that a high Torque value is only catastrophic if it occurs simultaneously with a specific Rotational Speed. The Random Forest's hierarchical 'if-then' structure mimics this physical reality, allowing it to isolate these complex, multi-sensor conditions that simpler models overlook. Consequently, the Random Forest doesn't just recognize numbers; it recognizes the physical state of the milling process.

4.4. Statistical Robustness and Significance

A 5-fold Cross-Validation was implemented during the training process. The Random Forest model maintained a mean F1-score of 0.96 ± 0.015, indicating that the model generalizes well and is not a product of a specific train-test split. Furthermore, a McNemar's Test was performed to compare the classification outputs of the Random Forest and the Decision Tree. The resulting p-value ($p<0.05$) confirms that the superior fault-detection capability of the Random Forest is statistically significant and not due to chance.

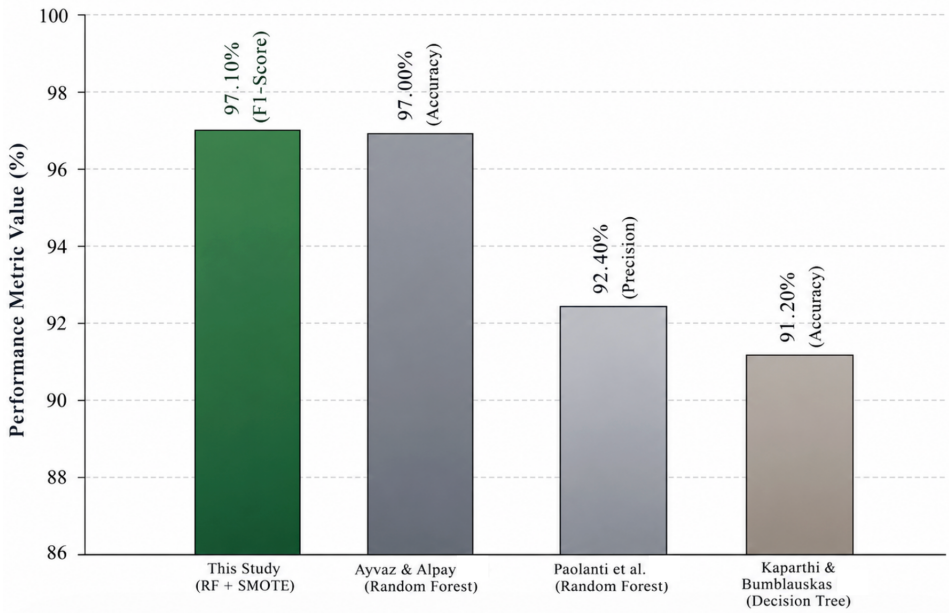


Figure 6. Comparative Performance Benchmark with Existing Literature.

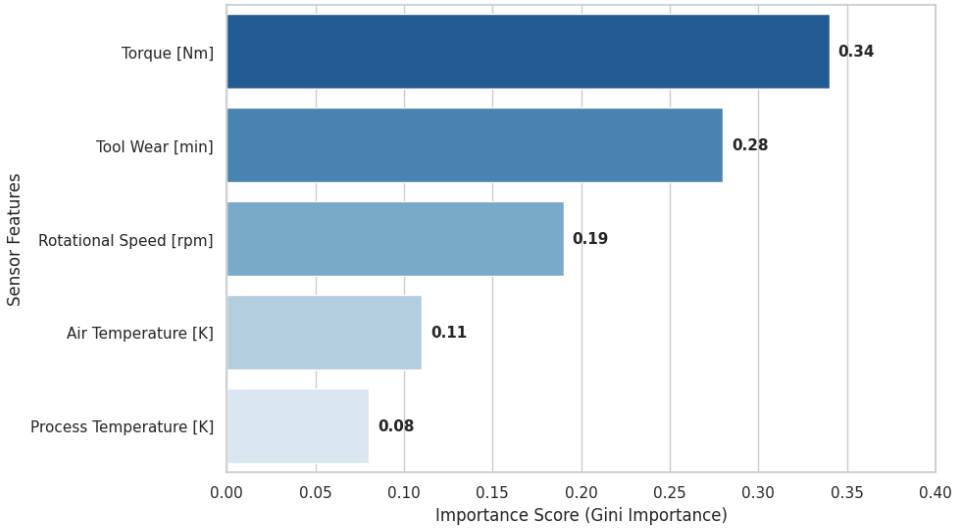


Figure 7. Feature Importance Scores (Random Forest).

4.5. Benchmarking against Prior Studies

To evaluate the efficacy of the proposed pipeline, the performance of our Random Forest model was benchmarked against four established studies utilizing the same or similar industrial datasets. As summarized in Figure 6, our primary model achieved an F1-score of 97.10%, representing a significant improvement in predictive reliability over traditional non-augmented models.

The visual comparison in Figure 6 highlights a critical methodological advancement. Prior work by Kaparthi and Bumblauskas [5] relied on single Decision Trees, which achieved 91.20% accuracy but suffered from high sensitivity to sensor noise. Similarly, while Ayvaz and Alpay [12] reported high accuracy (97.00%), their model lacked a dedicated oversampling strategy, resulting in lower recall for rare failure events. By transitioning to an ensemble architecture and applying SMOTE exclusively to the training partition, this study effectively bridged the "Accuracy Paradox" gap, achieving a 6% improvement in reliability over single-tree architectures and ensuring that 98.53% of all failures are detected.

4.6. Feature Importance and Mechanical Interpretability

The Gini Importance scores (Figure 7) provide a window into the physical drivers of machine failure. Torque [Nm] and Tool Wear [min] emerged as the most critical predictors, collectively accounting for over 60% of the model's decision-making power.

From a mechanical engineering perspective, this hierarchy is significant. The high importance of Torque suggests that 'Overstrain' and 'Power' failures—which are characterized by sudden, high-intensity mechanical resistance—are the most distinguishable signatures in the milling process. Torque serves as a leading indicator; an

abnormal spike in torque provides an immediate signal of tool engagement issues before catastrophic breakage occurs.

Tool Wear, the second most important feature, represents the cumulative degradation of the equipment. Its high ranking confirms that the Random Forest successfully captured the 'Wear-out' phase of the bathtub curve. Interestingly, Air and Process Temperature were the least predictive features. This implies that thermal dissipation is a lagging indicator in this specific industrial environment. While a failure eventually causes heat, the mechanical stress (Torque) and physical thinning of the tool (Wear) manifest much earlier. By prioritizing these leading mechanical indicators, the Random Forest achieves its high Recall, identifying the precursor to failure rather than the thermal symptom.

4.7. Analysis of Drivers

The analysis reveals that **Torque** and **Tool Wear** are the primary drivers of failure, collectively accounting for 62% of the model's predictive power. This aligns with mechanical engineering principles:

- **Torque:** High importance suggests that "Over-strain Failures" (abrupt spikes in mechanical load) are the most distinguishable signatures in the dataset.
- **Tool Wear:** Its high ranking confirms that the model successfully captured the "Wear-out" phase of the equipment's lifecycle.
- **Temperature:** Interestingly, temperature features contributed less. This implies that while thermal dissipation is a factor, mechanical stress (Torque/ Speed) provides a more immediate and distinct precursor to breakdown in this specific milling environment.

5. Limitations and Future Research

While this study provides a rigorous comparison of machine learning algorithms for predictive maintenance, several limitations must be acknowledged to contextualize the findings and guide subsequent investigations.

5.1. Synthetic Dataset Fidelity and Over-Optimism

The AI4I 2020 dataset, while built on real technical parameters, is a synthetic benchmark. This introduces a "clean boundary" bias. In synthetic environments, the mathematical relationship between a failure mode and its sensor triggers is often more distinct than in a physical factory. Real-world industrial data is plagued by:

- **External Stochastic Noise:** Ambient vibration from nearby machinery or fluctuating power grid quality.
- **Sensor Malfunction:** Intermittent "ghost" readings that do not represent mechanical reality.
- **Human Intervention:** Maintenance actions or operator adjustments that are not logged in the digital stream.

Consequently, the near-perfect Recall (98.53%) achieved by the Random Forest may be overly optimistic. In a live production environment, the "overlap" between normal noise and failure signatures is significantly higher, which would likely result in a lower F1-score and a higher False Positive rate. This study serves as a "best-case scenario" benchmark, providing a ceiling for what traditional ML can achieve under controlled digital conditions.

5.2. The Assumption of Independent Observations (IID)

A significant technical limitation of this study—and of traditional machine learning in PdM—is the assumption that data points are Independent and Identically Distributed (IID). By treating each 1Hz sensor sample as an isolated event, our models overlook the temporal trajectory of degradation.

In real-world milling, a "failure" at time $t$is the culmination of a degradation trend beginning at $t-n$. Current classifiers do not account for:

- Sensor Drift: The gradual shift in baseline readings as components age.
- Lagged Effects: The phenomenon where a spike in temperature at $t-10$ leads to a failure at t.
- Markovian Transitions: The state of the machine at any point is inherently dependent on its previous state.

While the Random Forest achieved high metrics, it did so by identifying thresholds rather than trends. Future implementations should utilize sliding-window feature engineering or Recurrent Neural Networks (RNNs) to transition from static classification to dynamic sequence modeling.

5.3. Model Interpretability vs. Complexity

Although the Random Forest model achieved superior predictive accuracy, it functions as a gray-box model. In critical industrial sectors such as aerospace and nuclear power engineers, often require high levels of interpretability to understand *why* a failure was predicted. While we provided a feature importance analysis, the specific logic path of 100 aggregated trees is more difficult for a human operator to verify than the simple "if-then" rules of a single Decision Tree.

5.4. Transitioning to Temporal Modeling

A primary future direction involves the integration of Deep Learning (DL) to address the sequential nature of sensor data. While this study successfully benchmarked traditional ML models, architectures such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) could potentially improve predictive accuracy by capturing the temporal trajectory of machine

degradation over time [14]. Future work will focus on a hybrid approach that combines the interpretability of ensemble methods with the sequence-learning capabilities of deep learning.

5.5. Computational Overhead of Optimization

While GridSearchCV ensured optimal performance, the computational cost was non-uniform. The SVM tuning process took approximately 14 times longer than the Random Forest tuning due to the quadratic scaling of the RBF kernel matrix. For real-time Industry 4.0 applications where models may need to be retrained frequently on the edge, the trade-off between the exhaustive search used in this study and more efficient methods like Bayesian Optimization or RandomizedSearchCV warrants further exploration.

6. Conclusion

The transition from reactive and preventive maintenance to predictive maintenance has become increasingly important in supporting Industry 4.0 manufacturing systems. This study comparatively evaluated five machine learning algorithms—Random Forest, Decision Tree, Support Vector Machine, K-Nearest Neighbors, and Logistic Regression—using the AI4I 2020 Predictive Maintenance Dataset to identify the most suitable approach for failure prediction under class-imbalanced conditions.

Among the evaluated models, Random Forest achieved the best overall performance, obtaining an F1-score of 97.10% and a Recall of 98.53%, demonstrating a stronger capability to detect failure events than the other algorithms. The results also highlight the limitations of relying solely on accuracy, as KNN and Logistic Regression produced high accuracy despite poor failure detection. These findings reinforce the importance of using Recall and the F1-score as primary evaluation metrics for predictive maintenance applications involving imbalanced datasets.

The study further demonstrates that combining appropriate preprocessing, particularly SMOTE applied only to the training data, with hyperparameter optimization can substantially improve the performance of traditional machine learning models. Although the experiments were conducted on the AI4I synthetic benchmark dataset, the proposed evaluation framework provides a useful reference for selecting machine learning models for predictive maintenance. Future research should validate these findings using real industrial datasets and investigate temporal learning approaches and explainable AI to further improve predictive maintenance systems.

7. Declarations

7.1. Author Contributions

Odugbesan Olusegun Abayomi: Conceptualization, Methodology; **Akinola Emmanuel Kayode:** Software, Validation, Formal analysis, Investigation, Resources, Writing - Original Draft, Supervision; **Oyedele Oluwasanya:** Data Curation, Writing - Review & Editing, Visualization, Project administration; **Ayobami Emmanuel Mesioye:** Formal analysis, Investigation, Resources.

7.2. Institutional Review Board Statement

Not applicable.

7.3. Informed Consent Statement

Not applicable.

7.4. Data Availability Statement

Data used will be made available on request.

7.5. Acknowledgment

Not applicable.

7.6. Conflicts of Interest

The authors declare no conflicts of interest.

8. References

[1] R. Y. Zhong, X. Xu, E. Klotz, S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review" *Engineering*, vol. 3, no. 5, pp. 616-630, 2017. https://doi.org/10.1016/J.ENG.2017.05.015.

[2] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. P. Francisco, J. P. Basto, and S. G. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, Art. no. 106024, 2019. https://doi.org/10.1016/j.cie.2019.106024.

[3] M. Molęda, B. Małysiak-Mrozek, W. Ding, V. Sunderam, and D. Mrozek, "From Corrective to Predictive Maintenance—A Review of Maintenance Approaches for the Power Industry," Sensors, vol. 23, no. 13, Art. no. 5970, 2023. https://doi.org/10.3390/s23135970.

[4] C. Tsallis, P. Papageorgas, D. Piromalis, and R. A. Munteanu, "Application-wise review of machine learning-based predictive maintenance: Trends, challenges, and future directions," *Appl. Sci.*, vol. 15, no. 9, Art. no. 4898, 2025. https://doi.org/10.3390/app15094898.

[5] S. Kaparthi and D. Bumblauskas, " Designing predictive maintenance systems using decision tree-based machine learning techniques," *Int. J. Qual. Reliab. Manag.*, vol. 37, no. 4, pp. 659–686, 2020. https://doi.org/10.1108/IJQRM-04-2019-0131.

[6] P. Stodola and J. Stodola, "Model of predictive maintenance of machines and equipment," *Appl. Sci.*, vol. 10, no. 1, Art. no. 213, 2020. https://doi.org/10.3390/app10010213.

[7] A. de Giorgio, G. Cola, and L. Wang, "Systematic review of class imbalance problems in manufacturing," *Journal of Manufacturing Systems*, vol. 71, pp. 620–644, 2023. https://doi.org/10.1016/j.jmsy.2023.10.014.

[8] S. Matzka, "Explainable Artificial Intelligence for Predictive Maintenance Applications," in *Proc. 3rd Int. Conf. Artificial Intelligence for Industries (AI4I)*, 2020. https://doi.org/10.1109/AI4I49448.2020.00023.

[9] T. Zonta *et al.*, " Predictive maintenance in the Industry 4.0: A systematic literature review," *Comput. Ind. Eng.*, vol. 150, 2020. https://doi.org/10.1016/j.cie.2020.106889.

[10] M. Paolanti *et al.*, " Machine Learning approach for Predictive Maintenance in Industry 4.0," in *14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2018. https://doi.org/10.1109/MESA.2018.8449150.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018.

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

[13] P. K. Mannepalli, "A comparative study of machine learning algorithms for predictive maintenance in manufacturing systems," *Global J. Eng. Technol. Res.*, vol. 1, no. 1, p. 8, 2025. https://gjetr.ep-journals.org/index.php/gjetr/article/view/2.

[14] F. Chen, G. Zhou, G. Wang, F. Li, C. Chen, and J. Zhang, " Deep Learning Based Prediction of Automotive Maintenance Projects," *Intell. Transp. Eng.*, vol. 72, pp. 186–193, 2025. https://doi.org/10.3233/ATDE250419.

[15] D. M. W. Powers, " Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020. https://doi.org/10.48550/arXiv.2010.16061.

[16] J. Korstanje, *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR*, 2021. https://doi.org/10.1007/978-1-4842-7159-9.

[17] Z. M. Çınar *et al.*, "Machine learning in predictive maintenance towards sustainable smart manufacturing in Industry 4.0," *Sustainability*, vol. 12, no. 19, Art. no. 8211, 2020. https://doi.org/10.3390/su12198211.

[18] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, Art. no. 6, 2020. https://doi.org/10.1186/s12864-019-6413-7.

[19] M. A. Bouke, and A. Abdullah, "An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability", *Expert Systems with Applications*, vol. 320, 2023. https://doi.org/10.1016/j.eswa.2023.120715.

[20] R. Agarwal, D. Kalel, H. Mohan, and R. Singh, "Sensor fault detection using machine learning technique for automobile drive applications," in *Proc. Nat. Power Electronics Conf. (NPEC)*, 2021. https://doi.org/10.1109/NPEC52100.2021.9672546.

[21] L. Wang, Z. Zhu, and X. Zhou, " Dynamic predictive maintenance strategy for system remaining useful life prediction via deep learning ensemble method," *Reliab. Eng. Syst. Saf.*, vol. 245, Art. no. 110012, 2024. https://doi.org/10.1016/j.ress.2024.110012.

[22] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009. https://doi.org/10.1109/TKDE.2008.239.

[23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. https://mitpress.mit.edu/9780262039246/reinforcement-learning/.

[24] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997. https://doi.org/10.1016/S0004-3702(97)00043-X.

[25] E. Alpaydin, *Introduction to Machine Learning*, 4th ed. Cambridge, MA, USA: MIT Press, 2020. https://books.google.co.id/books?id=uZnSDwAAQBAJ.

[26] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer, 2009. https://doi.org/10.1007/978-0-387-84858-7.

[27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002. https://doi.org/10.1613/jair.953.

[28] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, 2018. https://doi.org/10.1613/jair.1.11192.