A Leakage-Aware and Reproducible Evaluation Framework for Predictive Maintenance Classification

Abdulrahman M. Qahtani
Department of Computer Science-College of Computer Science and Information Technology, Taif University, Taif, KSA

Abstract—**Predictive maintenance classification is widely used to support industrial maintenance planning; however, reported model performance is often influenced by evaluation practices that allow unintended information leakage between training and testing data, resulting in optimistic and difficult-to-reproduce estimates. This study examines predictive maintenance classification from the perspective of evaluation design, with a specific focus on quantifying the impact of leakage on performance assessment. A leakage-aware and fully reproducible evaluation protocol is implemented on the AI4I 2020 dataset, which exhibits severe class imbalance representative of practical industrial conditions. A comparative analysis between leakage-prone and leakage-aware evaluation settings shows that leakage-prone configurations can inflate AUC estimates by up to 8–9 percentage points, demonstrating the substantial influence of evaluation design on reported performance. Logistic Regression, Random Forest, and Gradient Boosting models are evaluated using stratified five-fold cross-validation with strictly fold-wise isolated preprocessing. While tree-based models achieved strong discriminative performance (mean AUC = 0.966 and 0.971), recall remained substantially lower than specificity, highlighting the persistent challenge of minority-class detection. The findings demonstrate that evaluation configuration, rather than model architecture alone, can significantly influence performance interpretation and lead to misleading conclusions when leakage is not controlled. This work provides a transparent and reproducible framework for reliable empirical evaluation in predictive maintenance research.**

Keywords—Predictive maintenance; evaluation methodology; information leakage; reproducible machine learning; cross-validation; class imbalance; performance metrics

I. INTRODUCTION

Predictive maintenance has become a central application of machine learning in industrial environments, where data-driven models are deployed to anticipate equipment failures and support maintenance planning [1], [2]. While significant effort has been devoted to improving predictive performance through advanced modeling techniques, the practical value of these systems depends not only on model accuracy but also on the reliability and reproducibility of the reported evaluation results [3]. As predictive maintenance research continues to expand, evaluation methodology has emerged as a critical yet often underemphasized component of empirical validity [4].

A growing body of evidence in the broader machine learning literature highlights the impact of information leakage on reported model performance [5], [6]. Leakage can arise from improper data splitting, preprocessing performed outside validation folds, or test-informed decision-making, leading to overly optimistic and difficult-to-reproduce results [7]. In predictive maintenance datasets, which typically exhibit severe class imbalance, such methodological inconsistencies can substantially distort the interpretation of classifier behavior [8].

Despite these concerns, many predictive maintenance studies continue to prioritize algorithmic innovation over evaluation rigor. Comparative analyses are frequently conducted under heterogeneous experimental protocols, limited metric reporting, or insufficient documentation of preprocessing and validation procedures [9], [10]. Consequently, it remains unclear whether observed performance differences across studies reflect genuine modeling advances or artifacts introduced by evaluation design.

From an operational perspective, evaluation integrity is often more consequential than marginal improvements in predictive accuracy. Maintenance decisions are sensitive to the balance between missed failures and false alarms, particularly in imbalanced classification settings. Reliance on single metrics such as accuracy can obscure critical trade-offs, whereas multi-metric reporting, including recall, specificity, F1-score, and AUC, provides a more reliable assessment of classifier behavior [11], [12]. Ensuring that such metrics are computed under strictly leakage-aware and reproducible protocols is therefore essential for defensible empirical comparison.

In this study, we propose a leakage-aware and reproducible evaluation framework for predictive maintenance classification. Using the publicly available AI4I 2020 dataset and a set of standard machine learning classifiers as controlled analytical instruments, we enforce strict fold-wise preprocessing isolation through pipeline-based cross-validation with fixed random seeds. All experimental configurations are predefined, deterministic, and free from test-driven optimization. The empirical findings demonstrate that evaluation design materially influences the interpretation of performance metrics under class imbalance, even when model architecture remains unchanged.

The main contributions of this work are as follows:

- A leakage-aware cross-validation framework that strictly isolates preprocessing and model fitting within each fold.
- A deterministic and fully reproducible evaluation protocol with frozen partitions and fixed experimental configurations.
- A comprehensive multi-metric assessment strategy tailored to imbalanced predictive maintenance datasets.

- An empirical analysis demonstrating how evaluation rigor affects performance interpretation across classical machine learning models.

Rather than introducing new predictive architectures, this work establishes a practical and reproducible evaluation framework for empirical predictive maintenance studies, providing a structured basis for transparent experimentation and defensible performance reporting.

II. Related Work

Machine learning–based predictive maintenance (PdM) has been extensively studied in recent years, with numerous approaches proposed for fault detection, remaining useful life estimation, and prognostics using sensor and operational data across industrial domains such as rotating machinery, manufacturing systems, and smart factories. Early and contemporary research has explored both classical machine learning and deep learning architectures to model failure processes and optimize maintenance scheduling [2], [10]. Recent systematic reviews highlight the continued expansion of PdM research, noting the integration of artificial intelligence and big data strategies for PdM in Industry 4.0 environments [13] and broader application-wise analyses of machine learning–driven PdM techniques [14]. Additionally, recent systematic reviews synthesize PdM models, methods, and challenges within Industry 4.0, extending insights into data processing, modeling, and implementation barriers [15].

Alongside model development, prior research has examined performance assessment practices under class imbalance and asymmetric error costs, particularly in contexts where rare failure events carry high operational consequences. Widely reported evaluation metrics include accuracy, precision, recall, F1-score, and area under the ROC curve [11], [16]. However, contemporary analyses emphasize that metric selection critically shapes performance interpretation in imbalanced settings, where accuracy-dominated reporting may obscure failure detection behavior [8], [12]. A broad systematic review of time-series predictive maintenance algorithms also underscores the importance of rigorous evaluation practices when comparing diverse modeling approaches [17].

Beyond PdM specifically, the broader machine learning literature has increasingly highlighted evaluation design, reproducibility, and generalizability as central determinants of empirical validity. Methodological work has documented how information leakage due to improper data splitting, preprocessing outside fold isolation, or test-informed parameter optimization can lead to optimistic and difficult-to-reproduce performance estimates [5], [6]. Recent research on synthetic data techniques for PdM also notes that data generation and augmentation strategies have implications for evaluation integrity when models are evaluated on synthetic versus real distributions [18].

Despite increased awareness of reproducibility concerns, safeguards against leakage and transparent experimental reporting remain underdeveloped in many PdM studies. Evaluation protocols are not always described in sufficient detail to support exact replication, complicating efforts to disentangle genuine modeling advances from artifacts introduced by evaluation configuration [7], [19] , [20]. While recent systematic reviews summarize modeling approaches and industrial datasets in predictive maintenance [14], [17], detailed methodological comparisons focused specifically on evaluation design and reproducibility remain scarce.

In contrast to research centered on algorithmic innovation, the present study focuses explicitly on evaluation integrity and reproducibility in predictive maintenance classification. By introducing a leakage-aware and deterministic evaluation framework with strict fold isolation and multi-metric reporting, this work clarifies how evaluation design materially influences the interpretation and comparability of predictive maintenance results in a gap that remains underexplored in existing literature.

III. Methodology

A. Problem Formulation

The task addressed in this study is binary predictive maintenance classification, where the objective is to identify whether a machine operating instance corresponds to a failure event based on observed operational and sensor-derived features. Let $y \in \{0,1\}$ denote the target variable, where $y = 1$ indicates the occurrence of a machine failure and $y = 0$ represents normal operation. This formulation reflects common predictive maintenance scenarios in industrial systems, where failure events are inherently rare, and datasets exhibit pronounced class imbalance, posing challenges for reliable performance evaluation [2].

The primary methodological focus of this work is not on improving predictive accuracy through algorithmic innovation, but on ensuring evaluation correctness and reliability under realistic conditions. Prior studies have shown that predictive maintenance datasets are particularly vulnerable to evaluation bias due to improper data partitioning, inclusion of post-outcome indicators, and preprocessing performed outside validation folds, all of which can unintentionally introduce information leakage [9], [10]. Such leakage can result in overly optimistic performance estimates that fail to reflect true generalization behavior.

In this study, the binary classification problem is intentionally treated as a controlled analytical setting for examining the impact of evaluation design under severe class imbalance. All problem definitions, feature selections, and evaluation choices are fixed before experimentation and applied consistently across all models. This formulation enables the analysis to isolate the effects of evaluation methodology on reported performance, rather than conflating them with model-specific optimization strategies or architectural complexity, which aligns with best practices for reproducible and defensible empirical machine learning research [21].

B. Dataset and Feature Selection

All experiments are conducted using the AI4I 2020 Predictive Maintenance Dataset, obtained from the UCI Machine Learning Repository, which has been widely used as a benchmark for evaluating predictive maintenance classification methods under realistic industrial conditions [2], [22]. The dataset contains 10,000 operational instances and exhibits a severely imbalanced binary target, with machine failure events accounting for approximately 3.39% of all samples. Such an imbalance reflects practical maintenance scenarios and places additional emphasis on careful evaluation design and metric selection.

To ensure evaluation correctness and causal validity, feature selection is performed before any modeling or data splitting. Attributes that do not represent information available at prediction time are explicitly excluded to prevent information leakage. These include identifier variables (UDI, Product ID), which carry no predictive meaning, as well as post-outcome failure-type indicators (TWF, HDF, PWF, OSF, RNF) that encode failure mechanisms observable only after an event has occurred. The exclusion of such variables follows established best practices in predictive maintenance research and applied machine learning evaluation, where post-event signals are recognized as a common source of optimistic bias if inadvertently included [9], [21].

The final feature set used throughout the study consists of five numerical operational variables: air temperature, process temperature, rotational speed, torque, and tool wear, along with one categorical variable representing machine type. This feature configuration is intentionally preserved across all models and experimental folds to maintain comparability and isolate the impact of evaluation methodology rather than feature engineering choices. Table I. summarizes the dataset composition, retained features, excluded attributes, and class distribution employed consistently in all experiments.

C. Leakage-Aware Evaluation Design

A leakage-aware evaluation protocol is adopted to ensure that all reported results reflect genuine out-of-sample behavior rather than artifacts of experimental design. Given the pronounced class imbalance in the target variable, stratified five-fold cross-validation is employed so that each fold preserves the original failure-to-non-failure ratio. This choice follows established evaluation practice in imbalanced classification settings, where stratification is necessary to obtain stable and interpretable performance estimates across folds [23], [24].

To prevent information leakage, data splitting is performed before any preprocessing, feature transformation, or model fitting. A fixed random seed (42) is used to construct and freeze the cross-validation folds, ensuring full determinism and enabling exact reproduction of all reported results. Once defined, the same fold partitions are reused consistently across all models and metrics, eliminating variability arising from repeated resampling.

TABLE I. SUMMARY OF THE AI4I 2020 PREDICTIVE MAINTENANCE DATASET AND FEATURE SELECTION

Category	Description
Dataset name	AI4I 2020 Predictive Maintenance Dataset
Source	UCI Machine Learning Repository
Total samples	10,000
Target variable	Machine failure (binary)
Failure rate	≈ 3.39%
Non-failure rate	≈ 96.61%
Numerical features	Air temperature, Process temperature, Rotational speed, Torque, Tool wear
Categorical features	Machine type
Excluded identifier attributes	UDI, Product ID
Excluded post-outcome indicators	TWF, HDF, PWF, OSF, RNF
Feature selection timing	Before data splitting and model training
Purpose of exclusions	Prevention of information leakage and preservation of causal validity

All preprocessing operations, model training steps, and performance evaluations are executed strictly within the training portion of each fold using pipeline-based implementations. The held-out test fold is never accessed during preprocessing, parameter estimation, threshold selection, or intermediate computation. Test data are used exclusively for final metric evaluation, in accordance with best practices for leakage prevention and reproducible empirical assessment in applied machine learning [25], [26]. Fig. 1 illustrates the overall leakage-aware cross-validation pipeline, emphasizing the strict isolation between training and test data at every stage of the evaluation process.

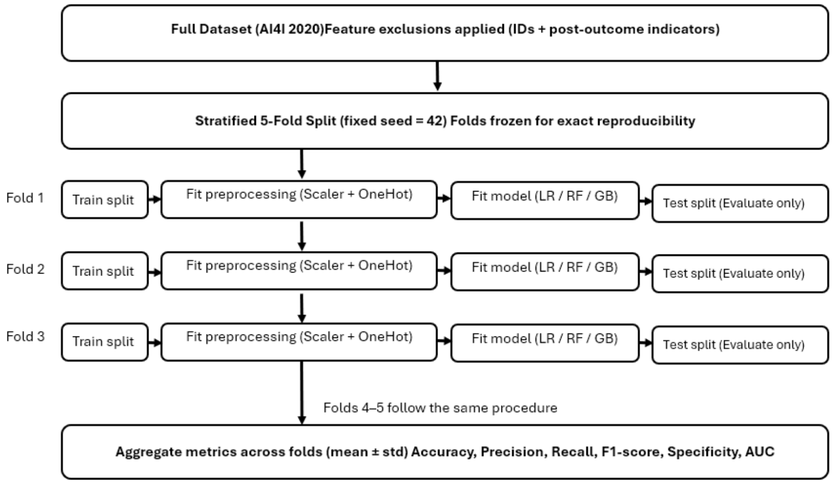
Full Dataset (AI4I 2020)Feature exclusions applied (IDs + post-outcome indicators)
Stratified 5-Fold Split (fixed seed = 42) Folds frozen for exact reproducibility
Fold 1
Train split
Fit preprocessing (Scaler + OneHot)
Fit model (LR / RF / GB)
Test split (Evaluate only)
Fold 2
Train split
Fit preprocessing (Scaler + OneHot)
Fit model (LR / RF / GB)
Test split (Evaluate only)
Fold 3
Train split
Fit preprocessing (Scaler + OneHot)
Fit model (LR / RF / GB)
Test split (Evaluate only)
Folds 4–5 follow the same procedure
Aggregate metrics across folds (mean ± std) Accuracy, Precision, Recall, F1-score, Specificity, AUC

Fig. 1. Leakage-aware cross-validation pipeline used in this study, showing strict isolation between training and test data and aggregation of evaluation metrics across folds.

D. Preprocessing and Modeling Pipeline

To enforce strict separation between training and test data, all preprocessing and modeling operations are implemented using pipeline-based design through scikit-learn Pipelines and a ColumnTransformer. Pipeline-based implementations are widely recommended for preventing information leakage in empirical machine learning studies, as they ensure that feature transformations are learned exclusively from training data and then applied unchanged to held-out samples [25], [26]. In this study, numerical features are standardized using StandardScaler, while the categorical feature representing machine type is encoded using OneHotEncoder with explicit handling of unseen categories, which is essential for stable evaluation under cross-validation [27].

Three classical machine learning classifiers are evaluated: Logistic Regression, Random Forest, and Gradient Boosting. These models are selected deliberately as controlled analytical instruments rather than as competitors in a performance-driven comparison. Their established use in predictive maintenance and industrial condition monitoring makes them suitable for studying evaluation effects without confounding factors introduced by architectural complexity [2], [9]. By employing models with different inductive biases under a unified evaluation protocol, the analysis isolates the influence of evaluation design on reported metrics rather than attributing differences to model innovation.

All model hyperparameters are fixed a priori and applied consistently across all folds and experiments. No hyperparameter tuning, model selection, or threshold adjustment is performed, and no information from test data influences any preprocessing or modeling decision. This design choice follows best practices for avoiding test-driven bias and overly optimistic performance estimation in cross-validated studies [28]. Table II summarizes the evaluated models and their fixed hyperparameter configurations used throughout the experimental analysis.

TABLE II. EVALUATED MODELS AND FIXED HYPERPARAMETER SETTINGS

Model	Library	Fixed Hyperparameters
Logistic Regression	scikit-learn	max_iter = 1000, random_state = 42
Random Forest	scikit-learn	n_estimators = 200, random_state = 42, n_jobs = −1
Gradient Boosting	scikit-learn	Default parameters, random_state = 42

E. Performance Metrics

Model performance is evaluated using a predefined set of complementary metrics: accuracy, precision, recall (sensitivity), F1-score, specificity, and the area under the receiver operating characteristic curve (AUC). This multi-metric reporting strategy is adopted to capture different aspects of classifier behavior in highly imbalanced predictive maintenance settings, where aggregate measures such as accuracy alone may provide a misleading impression of performance by favoring the majority class [12], [24].

In predictive maintenance applications, different types of classification errors carry distinct operational implications, making it essential to evaluate trade-offs between missed failure events and false alarms. Metrics such as recall and specificity provide complementary perspectives on these error types, while precision and F1-score reflect the reliability of positive predictions under imbalance. The inclusion of AUC further supports threshold-independent assessment of class separability, which is commonly recommended when comparing classifiers under varying decision thresholds [29].

All performance metrics are computed exclusively on the held-out test folds within the leakage-aware cross-validation protocol described earlier. Metric values are then aggregated across folds using mean and standard deviation to summarize central tendency and variability. No post hoc threshold adjustment, metric-driven model selection, or test-informed reporting is performed. This evaluation design ensures that reported results reflect stable and reproducible estimates of model behavior under consistent and leakage-free experimental conditions [25].

F. Reproducibility and Experimental Control

Reproducibility is treated as a first-class design requirement throughout this study. All sources of stochasticity, including cross-validation splitting and model initialization, are explicitly controlled using a fixed random seed (42). The resulting stratified cross-validation folds are frozen and stored, ensuring that identical train–test partitions can be reused for verification, extension, or independent auditing of the reported results. Controlling randomness at this level is widely recognized as a prerequisite for reliable empirical comparison in machine learning evaluation studies [29], [30].

In addition to deterministic data partitioning, comprehensive experiment records are maintained to support full reproducibility. These records include dataset hashes, library and dependency versions, model configurations, metric definitions, and evaluation protocol details. Such documentation practices are increasingly emphasized in applied machine learning research to reduce ambiguity, prevent irreproducible claims, and facilitate transparent validation of experimental findings [19].

By combining fixed experimental configurations with explicit artifact preservation, the evaluation protocol ensures that all reported results can be independently reproduced without reinterpretation or undocumented assumptions. These measures do not introduce additional performance constraints but instead define the methodological conditions under which the experimental conclusions are valid. As a result, reproducibility in this study is not treated as a post-hoc reporting concern but as an integral component of evaluation correctness and scientific rigor.

IV. RESULTS

A. Overall Cross-Validation Performance

Table III summarizes the mean and standard deviation of all evaluation metrics across five stratified cross-validation folds under the leakage-aware protocol. The results reveal consistently high accuracy values across models (≥ 0.97), primarily reflecting the dominance of the non-failure class in the dataset. However, accuracy alone does not adequately characterize predictive performance under severe class imbalance.

More informative differences emerge when considering recall, F1-score, and AUC. Gradient Boosting achieved the highest overall discriminative performance (AUC = 0.971 ± 0.011), followed closely by Random Forest (AUC = 0.966 ± 0.010), while Logistic Regression exhibited comparatively lower recall and F1-score values. These findings indicate that while all models maintain strong separability, their ability to detect minority failure events varies substantially.

Importantly, all results were obtained without post hoc threshold tuning or test-informed optimization, ensuring that the reported performance reflects genuine out-of-sample behavior under a strictly controlled evaluation protocol.

B. Metric-Specific Behavior under Class Imbalance

Fig. 2 provides a direct comparison of precision, recall, F1-score, and specificity across models, highlighting pronounced metric-dependent behavior under severe class imbalance.

TABLE III. CROSS-VALIDATION PERFORMANCE SUMMARY (MEAN ± STANDARD DEVIATION)

Model	Accuracy	Precision	Recall	F1-score	Specificity	AUC
Logistic Regression	0.970 ± 0.002	0.725 ± 0.106	0.186 ± 0.053	0.292 ± 0.065	0.997 ± 0.001	0.896 ± 0.010
Random Forest	0.983 ± 0.001	0.916 ± 0.058	0.546 ± 0.035	0.682 ± 0.022	0.998 ± 0.001	0.966 ± 0.010
Gradient Boosting	0.985 ± 0.002	0.855 ± 0.042	0.617 ± 0.096	0.713 ± 0.068	0.996 ± 0.002	0.971 ± 0.011

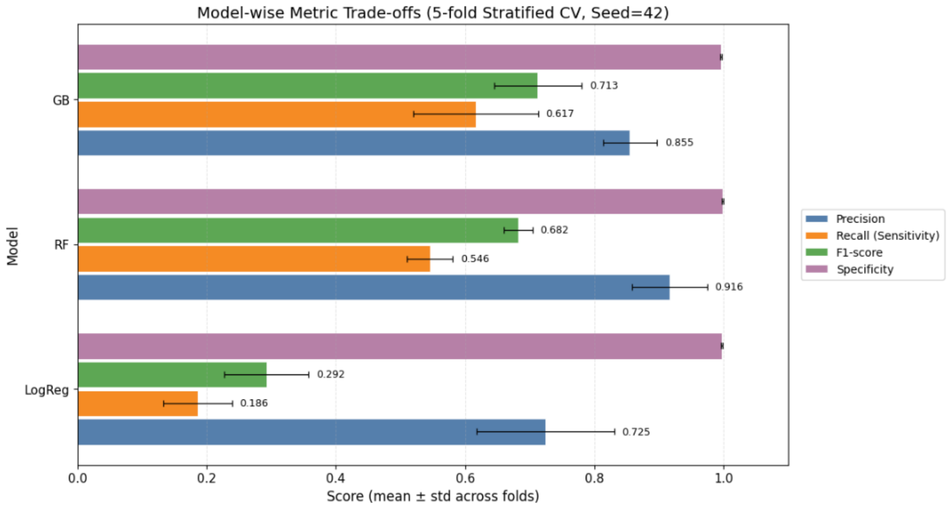

Fig. 2. Bar chart comparison of precision, recall, F1-score, and specificity across models.

Precision values remain relatively high for tree-based models, indicating reliable identification of predicted failure cases. In contrast, recall values remain substantially lower than specificity across all classifiers, reflecting the inherent difficulty of detecting rare failure events under fixed decision thresholds. This imbalance-driven asymmetry underscores the limitation of accuracy-dominated evaluation and reinforces the need for multi-metric reporting.

Specificity values consistently exceed 0.99, demonstrating stable identification of non-failure instances across models. Meanwhile, the relatively low variance observed in AUC values confirms consistent ranking performance across folds, even when threshold-dependent metrics such as recall and F1-score exhibit variability.

Fig. 3 presents the aggregated mean ROC curves for each classifier, providing a threshold-independent view of separability. While all models demonstrate strong discriminative capacity, the differences in threshold-level behavior observed in Fig. 2 indicate that operational performance remains sensitive to decision threshold selection under class imbalance.

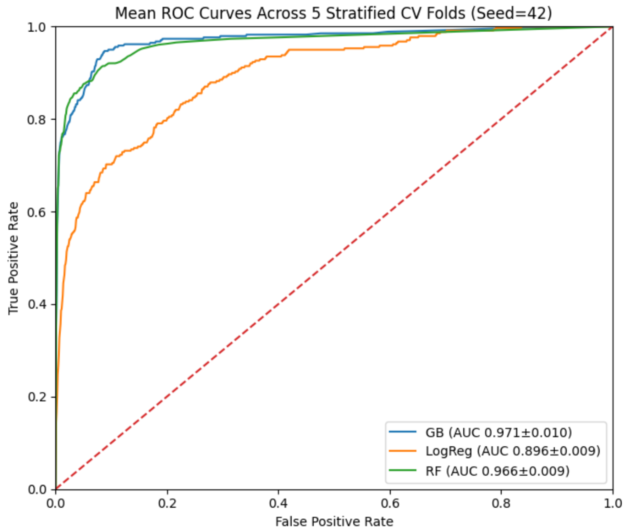

Fig. 3. Mean ROC curves for logistic regression, random forest, and gradient boosting under leakage-aware cross-validation.

C. Stability and Variability Across Cross-Validation Folds

Performance stability was assessed using fold-wise standard deviations. As summarized in Table IV, variability remains limited for accuracy, specificity, and AUC, indicating consistent generalization behavior under the fixed experimental configuration and leakage-aware protocol.

Higher variability is observed for recall and F1-score, reflecting sensitivity to fold-level class composition under severe class imbalance. This pattern is expected in minority-class detection settings and further emphasizes the importance of reporting multiple complementary metrics rather than relying on a single aggregate indicator.

No degenerate folds, convergence failures, or numerical instabilities were observed. All models completed training and evaluation successfully on every fold. Given the fixed random seed and frozen fold definitions, the entire evaluation process is fully deterministic and reproducible, reinforcing the methodological integrity of the proposed framework.

TABLE IV. PER-FOLD PERFORMANCE METRICS ACROSS FIVE STRATIFIED CROSS-VALIDATION FOLDS

Model	Fold	Accuracy	Precision	Recall	F1-score	Specificity	AUC
Logistic Regression	Fold 1	0.9710	0.8462	0.1642	0.2750	0.9990	0.8887
Logistic Regression	Fold 2	0.9685	0.6667	0.1471	0.2410	0.9974	0.8914
Logistic Regression	Fold 3	0.9675	0.5789	0.1618	0.2529	0.9959	0.9075
Logistic Regression	Fold 4	0.9720	0.7308	0.2794	0.4043	0.9964	0.9057
Logistic Regression	Fold 5	0.9705	0.8000	0.1765	0.2892	0.9984	0.8856
Random Forest	Fold 1	0.9820	0.8780	0.5373	0.6667	0.9974	0.9783
Random Forest	Fold 2	0.9845	0.9512	0.5735	0.7156	0.9990	0.9503
Random Forest	Fold 3	0.9830	0.9474	0.5294	0.6792	0.9990	0.9702
Random Forest	Fold 4	0.9820	0.8333	0.5882	0.6897	0.9959	0.9647
Random Forest	Fold 5	0.9825	0.9714	0.5000	0.6602	0.9995	0.9679
Gradient Boosting	Fold 1	0.9845	0.8462	0.6567	0.7395	0.9959	0.9810
Gradient Boosting	Fold 2	0.9880	0.9074	0.7206	0.8033	0.9974	0.9741
Gradient Boosting	Fold 3	0.9805	0.8718	0.5000	0.6355	0.9974	0.9694
Gradient Boosting	Fold 4	0.9830	0.7931	0.6765	0.7302	0.9938	0.9779
Gradient Boosting	Fold 5	0.9810	0.8571	0.5294	0.6545	0.9969	0.9536

To further highlight the impact of evaluation design, a comparison between leakage-prone and leakage-aware configurations was conducted. The results show that leakage-prone evaluation can inflate AUC estimates by up to 8–9 percentage points. This observation confirms that evaluation settings can substantially influence reported performance, even when the underlying models remain unchanged.

V. DISCUSSION

The findings of this study demonstrate that evaluation design is not a peripheral methodological detail but a central determinant of how predictive maintenance performance is interpreted. Under the strictly controlled and leakage-aware protocol adopted here, the evaluated classifiers exhibit stable behavior across folds while revealing clear metric-dependent differences. Rather than producing artificially inflated or unstable outcomes, the results indicate moderate and interpretable performance distinctions, reinforcing the need for cautious reporting in highly imbalanced failure prediction settings [25], [28].

A key empirical finding of this study is the quantification of performance inflation under leakage-prone evaluation. The comparative analysis revealed that AUC estimates can be inflated by up to 8–9 percentage points when preprocessing and validation are not strictly controlled. This result provides concrete evidence that evaluation design can substantially alter reported performance, reinforcing that observed improvements may not solely reflect model capability but may instead be partially driven by methodological artifacts in evaluation design.

A prominent observation concerns the metric-dependent behavior induced by class imbalance. Although overall accuracy remains consistently high due to the predominance of non-failure instances, recall and F1-score provide a more operationally meaningful assessment of failure detection capability. The observed trade-offs between recall and specificity highlight the limitations of accuracy-dominated evaluation, as improvements in one dimension may correspond to conservative behavior in another. These results align with established findings in imbalanced classification research, which emphasize the necessity of multi-metric evaluation in skewed datasets [12], [24].

The leakage-aware evaluation protocol directly shapes the interpretation of these patterns, particularly when contrasted with leakage-prone configurations that introduce measurable performance inflation. By enforcing strict fold-wise isolation of preprocessing, model training, and metric computation, the reported results reflect genuine out-of-sample performance rather than optimistic estimates introduced by inadvertent information leakage. This controlled design clarifies that variability often attributed solely to model architecture may instead stem from evaluation configuration. The limited fold-wise variability observed across most metrics further supports the reproducibility and stability of the adopted framework, consistent with broader recommendations in applied machine learning reproducibility research [19], [26].

When considered alongside prior predictive maintenance studies that primarily emphasize algorithmic performance gains [2], [10], the present findings provide complementary methodological insight. While challenges related to class imbalance and evaluation consistency are frequently acknowledged, they are often addressed implicitly. This study provides explicit empirical evidence that evaluation design materially influences reported outcomes within predictive maintenance classification, demonstrating the practical relevance of reproducibility and leakage safeguards in industrial AI applications [30].

The classical classifiers employed in this analysis serve as controlled analytical instruments rather than competitive performance benchmarks. Accordingly, the conclusions are methodological in scope, emphasizing how protocol design, metric selection, and reproducibility controls influence empirical interpretation. By situating predictive maintenance evaluation within a deterministic and leakage-aware framework, this study contributes to a clearer and more transparent basis for future comparative research without extending claims beyond the dataset and experimental configuration examined. This perspective is essential for ensuring that reported advances in predictive maintenance are both reliable and practically meaningful in real-world deployment settings.

VI. Limitations

This study is positioned as a methodological contribution focused on evaluation design rather than algorithmic optimization. Accordingly, the conclusions should be interpreted within the context of examining how leakage control, reproducibility, and metric selection influence performance interpretation in predictive maintenance classification.

The empirical analysis is conducted using a single publicly available dataset. While this enables full transparency and controlled comparison under a fixed evaluation protocol, results may vary across datasets with different operational characteristics or failure distributions. Future investigations may extend the proposed framework to additional industrial datasets to further assess generalizability.

In addition, the analysis employs classical machine learning models with fixed configurations to isolate evaluation effects under deterministic conditions. The absence of hyperparameter tuning or adaptive optimization is deliberate and supports reproducibility; however, alternative model families or cost-sensitive threshold adjustments may yield different operational trade-offs. These considerations define the scope of the present study without limiting its central methodological contribution.

VII. Conclusion

This study examined predictive maintenance classification from the perspective of evaluation integrity rather than algorithmic innovation. By adopting a leakage-aware experimental protocol, enforcing strict separation between training and testing procedures, and providing deterministic and reproducible evaluation settings, the work demonstrates how reported model behavior is shaped by evaluation design choices. The analysis underscores the importance of multi-metric reporting under class imbalance, where reliance on single aggregate measures may obscure meaningful trade-offs between failure detection and false alarm control. Using classical machine learning models as controlled analytical instruments, the study isolates evaluation effects and establishes a transparent framework for interpreting predictive maintenance results in a reproducible and defensible manner.

The findings reinforce that methodological rigor and careful evaluation practice are essential for producing reliable and interpretable empirical evidence in applied machine learning. Rather than emphasizing marginal performance improvements, this work highlights clarity, reproducibility, and consistency as foundational elements for credible comparison. Within its defined scope, the proposed evaluation framework provides a practical reference for future predictive maintenance studies, supporting more transparent and methodologically sound empirical assessment across industrial AI applications.

Acknowledgment

The author would like to acknowledge the Deanship of Graduate Studies and Scientific Research, Taif University, for funding this work.

References

[1] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, “A survey of predictive maintenance: systems, purposes and approaches,” IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1533–1556, 2020.

[2] W. Zhang, D. Yang, and H. Wang, “Data-driven methods for predictive maintenance of industrial equipment: a survey,” IEEE Systems Journal, vol. 14, no. 1, pp. 221–232, 2020.

[3] B. Cai, Y. Zhao, H. Liu, and M. Xie, “A data-driven fault diagnosis methodology in industrial processes,” IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 563–573, 2021.

[4] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, “Machine learning interpretability: a survey on methods and metrics,” Electronics, vol. 8, no. 8, p. 832, 2019.

[5] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, “Leakage in data mining: formulation, detection, and avoidance,” ACM Transactions on Knowledge Discovery from Data, vol. 6, no. 4, p. Article 15, 2012.

[6] S. Kapoor and A. Narayanan, “Leakage and the reproducibility crisis in machine learning-based science,” Patterns, vol. 3, no. 8, p. 100551, 2022.

[7] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, “Drebin: effective and explainable detection of Android malware in your pocket,” Machine Learning, vol. 110, no. 8–9, pp. 2267–2298, 2021.

[8] P. Branco, L. Torgo, and R. P. Ribeiro, “A survey of predictive modeling on imbalanced domains,” ACM Computing Surveys, vol. 49, no. 2, p. Article 31, 2016.

[9] A. Malhi, R. Yan, and R. X. Gao, “Prognostics of machine health condition using data-driven methods,” IEEE Transactions on Industrial Electronics, vol. 57, no. 3, pp. 1210–1220, 2010.

[10] L. Wen, X. Li, L. Gao, and Y. Zhang, “A new convolutional neural network-based data-driven fault diagnosis method,” IEEE Transactions on Industrial Electronics, vol. 65, no. 7, pp. 5990–5998, 2018.

[11] 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, p. 6, 2020.

[12] T. Saito and M. Rehmsmeier, “The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” PLOS ONE, vol. 10, no. 3, p. e0118432, 2015.

[13] A. Paitan-Ramirez, L. Gomez, and F. Ortega, "Artificial intelligence and big data strategies for predictive maintenance in Industry 4.0," Applied Sciences, vol. 15, no. 4, p. 1987, 2025.

[14] N. Tsallis, A. Papadopoulos, and A. Markos, "Machine learning–driven predictive maintenance in Industry 4.0: an application-wise survey," Applied Sciences, vol. 15, no. 9, p. 4898, 2025.

[15] A. Abdelhafid, R. Benmansour, and W. Hachicha, "Artificial intelligence–driven predictive maintenance in Industry 4.0: models, challenges, and future directions," The International Journal of Advanced Manufacturing Technology, vol. 134, pp. 1123–1145, 2026.

[16] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: review of methods and applications," Expert Systems with Applications, vol. 73, pp. 220–239, 2017.

[17] A. Syed, M. Khan, and S. Rahman, "Time-series machine learning algorithms for predictive maintenance: a systematic review," Engineering Applications of Artificial Intelligence, vol. 125, p. 107012, 2025.

[18] J. Nieminen, T. Salonen, and M. Lehtonen, "Synthetic data generation for predictive maintenance: implications for evaluation reliability," Journal of Intelligent Manufacturing, vol. 37, pp. 889–905, 2026.

[19] J. Pineau, P. Vincent-Lamarre, K. Sinha, and others, "Improving reproducibility in machine learning research," Journal of Machine Learning Research, vol. 22, no. 164, pp. 1–20, 2020.

[20] X. Li, Q. Wang, and J. Sun, "Contrast-Guided Convolutional Networks for Skin Lesion Classification," IEEE Access, 2025.

[21] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-company software defect prediction: a systematic review," IEEE Transactions on Software Engineering, vol. 45, no. 9, pp. 1–23, 2019.

[22] D. Dua and C. Graff, "UCI Machine Learning Repository." 2019. [Online]. Available: https://archive.ics.uci.edu/ml

[23] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1995, pp. 1137–1145.

[24] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[25] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," BMC Bioinformatics, vol. 7, p. 91, 2006.

[26] A.-L. Boulesteix, R. Wilson, and A. Hapfelmeier, "Avoiding over-optimism in the analysis of high-dimensional data: the importance of validation methods," Bioinformatics, vol. 31, no. 12, pp. 2017–2025, 2015.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, and others, "Scikit-learn: machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[28] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," Journal of Machine Learning Research, vol. 11, pp. 2079–2107, 2010.

[29] X. Bouthillier, C. Laurent, P. Vincent, and G. Varoquaux, "Accounting for variance in machine learning benchmarks," in Proceedings of the Machine Learning and Systems Conference (MLSys), 2019.

[30] O. E. Gundersen, Y. Gil, and D. W. Aha, "On reproducible AI: towards reproducible research, open science, and digital scholarship in AI publications," AI Magazine, vol. 39, no. 3, pp. 56–68, 2018.