SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

A Hybrid Stacked-Ensemble Deep Learning Framework for Machine Learning-Based Predictive Maintenance in Smart Manufacturing Operations

Dr. Prajali Valmik Patil[1], Ms. Smita Shivaji Chavan[2], Dr. Sachin Ramchandra Misal[3], Dr. Gorakh Laxman Wakhare[4], Dr. Prashant Bhavarlal Chordiya[5], Prakash V. Sontakke[6]

[1]Assistant Professor, Department of Master of Computer Applications Dr. D Y Patil Center For Management & Research,Chikhali, Pune, India. E-mail: prajalivpatil@gmail.com, ORCID: 0009-0003-7068-4452
[2]Assistant Professor, Department of Master of Computer Applications Suryadatta Institute of Business Management and Technology, Pune, India. E-mail: smita.smitha@gmail.com, ORCID: 0009-0004-2654-007X
[3]Assistant Professor, Department of Master of Business AdministrationInternational Institute of Management Science (IIMS), Chinchwad, Pune, India. E-mail: misal.sachin@gmail.com, ORCID: 0000-0002-7373-6508
[4]Professor Department of MBA D.Y. Patil Institute of Master of Computer Applications and Management, Akurdi, Pune, India.
E-mail: wakhare@gmail.com
[5]Associate Professor, Department of Master of Computer Applications D.Y. Patil Institute of Master of Computer Applications and Management, Akurdi, Pune, India. E-mail: prashant.chordiya@gmail.com, ORCID: 0009-0005-6227-2583
[6]Assistant Professor Department of Electronics and Telecommunications Pimpri Chinchwad College of Engineering, Pune, India.
E-mail: prakash.sontakke@pccoepune.org, ORCID: 0000-0002-7129-7513

Abstract
Unplanned equipment downtime remains one of the most persistent sources of productivity loss in discrete and process manufacturing, with industry surveys attributing substantial annual revenue loss to reactive maintenance practices. While machine learning (ML) has been widely proposed as a substitute for time-based and reactive maintenance strategies, most published predictive maintenance (PdM) studies rely on a single learning algorithm and report performance on heavily engineered, often near-noise-free benchmarks, leaving open questions about robustness under class imbalance and about the transparency of model decisions to maintenance engineers. This study proposes SE-PdM, a hybrid stacked-ensemble framework that combines two heterogeneous base learners — a Random Forest (RF) classifier and an Extreme Gradient Boosting (XGBoost) classifier — with a one-dimensional Convolutional Neural Network coupled to a Bidirectional Long Short-Term Memory network (1D-CNN-BiLSTM) operating on sliding-window sensor sequences, integrated through a logistic-regression meta-learner. A strict class imbalance of failure modes is dealt with through a hybrid SMOTE-Tomek resampling strategy within training folds while a SHAP value is computed to explain the decisions of the ensemble model to the plant engineers. The framework is tested on the AI4I 2020 predictive maintenance benchmark (10,000 operating records, five failure modes), stratified 5-fold cross validation. Experimental results indicate that SE-PdM is superior to all base learners and to the following deep-learning based baselines for both macro-averaged F1-score and ROC-AUC and that the most important features for the prediction of failure risk are always torque, tool wear and the interaction between the rotational speed and the process temperature, irrespective of the base learner. This paper makes the following three key contributions: (i) a tabular sensor data representation that incorporates a tree-based and sequence-based representation; (ii) a training protocol that does not suffer from resampling leakage between cross validation folds, a common, but underreported pitfall in past PdM research; (iii) an interpretability layer that maps model output into actionable maintenance triggers. The proposed model provides an operational example of explainable predictive maintenance pipelines in smart manufacturing with the consideration of imbalance.

Keywords: Predictive Maintenance; Smart Manufacturing; Ensemble Learning; Deep Learning; Explainable AI; Industry 4.0; Class Imbalance; SHAP

1. Introduction

The transition from traditional factory floors to cyber-physical, sensor-instrumented production systems — commonly framed under the Industry 4.0 paradigm — has fundamentally changed how manufacturers manage the reliability of their assets. Modern machine tools, motors, and process lines are now instrumented with vibration, temperature, torque, and acoustic sensors that stream continuous condition data to edge and cloud platforms. This data abundance has created the technical precondition for predictive maintenance (PdM): the practice of estimating the future health state of an asset from historical and real-time operating data so that

maintenance can be scheduled just before failure, rather than on a fixed calendar (time-based maintenance) or after breakdown (reactive maintenance).

Reactive and preventive maintenance strategies remain dominant in many manufacturing sectors despite their well-documented inefficiencies: reactive maintenance incurs unplanned downtime and secondary damage, while preventive maintenance often replaces components well before the end of their useful life, wasting spare-part budget and labor. Machine learning-based PdM promises to close this gap by learning the statistical relationship between sensor signatures and impending failure, enabling condition-based interventions. However, the literature reviewed in Section 5 shows that most PdM studies (a) evaluate a single algorithm in isolation rather than a principled ensemble, (b) report accuracy without accounting for the severe class imbalance inherent to failure data, where healthy operating records vastly outnumber failure records, and (c) provide limited insight into which sensor variables drive individual predictions — a critical gap for maintenance engineers who must trust and act on model output.

This study directly tackles these three gaps. The main research question is whether a multi-class failure prediction system that integrates tabular feature interactions (tabular learners) with short horizon temporal dynamics (recurrent-convolutional learner) can be better than any single component prediction system while maintaining the interpretability of the models and their insensitivity to class imbalance under a leakage-free evaluation protocol. To this end, the present work aims to achieve three goals: (1) Design a stacked-ensemble architecture (SE-PdM), which combines Random Forest, XGBoost and a 1D-CNN-BiLSTM base learner with a meta-learner; (2) Implement and empirically validate a resampling protocol that does not suffer from the fold-leakage problem as seen in a subset of previous AI4I 2020 works; (3) Integrate SHAP-based explainability so that model output can be translated into maintenance recommendations that are actionable.

The summary of the contributions of this paper is given below. First, it brings a stacking architecture that the authors feel has not been studied so far in this specific stacking combination; and second, they report a measurable improvement in macro-F1 and ROC-AUC on the AI4I 2020 data compared to individual base learners. Secondly, it explicitly illustrates and statistically proves that resampling using SMOTE-Tomek on training folds impacts the performance, whereas on the test folds it gives optimistic results, being resampled beforehand of the cross validation splitting. Third, it translates explainability into actionable maintenance steps, through SHAP feature-attribution plots, connecting the dots between model accuracy and engineering use. In Section 5 recent literature review and positioning of the present contribution is provided, description of the proposed methodology is given in Section 6, description of the dataset and pre-processing pipeline is provided in Section 7, experimental setup is outlined in Section 8, results are provided and discussed in Section 9 and conclusion and future work is outlined in Section 10 and 11 respectively.

2. Literature Review

Research on predictive maintenance has progressed from feasibility studies on a single model to systematic reviews, benchmarking and explainability as a way of standardizing the model in the past 5 years. Carvalho et al. (2019) performed a preliminary systematic literature review on machine learning techniques for PM, summarizing the most common use of Random Forest, Support Vector Machines, and Artificial Neural Networks as the algorithms most commonly used for PM, and noting the lack of publicly available realistic failure datasets as a key factor that impedes repeatable research. On this synthesized work, Zonta et al. (2020) conducted a systematic review of the PdM research in the broader technology stack of Industry 4.0, detecting that most of the current papers published are related to the laboratory testing of the technology and not to its application on the shop floor.

A second wave of research has concentrated on comparative benchmarking in terms of the AI4I 2020 synthetic predictive maintenance dataset explicitly created by Matzka (2020) to be statistically representative of the structure of real-life telemetry from a milling-machine, but which is still publicly redistributable. On this benchmark, a few independent works have demonstrated the performance of ensemble-based tree learners to be consistently superior to linear and instance-based classifiers, with macro-F1 scores around 0.88 for ensemble-based Random Forest and 0.90 for XGBoost classifiers. This is illustrated by a recent comparison of imbalance handling techniques on the same dataset in which no resampling techniques were used and even the best baseline model XGBoost achieved zero recall for the rarest failure subtype (tool wear failure) when it was highly accurate on the majority of the other failure subtypes.

The remaining-useful-life (RUL) estimation and fault classification are also crucial problems that have been studied using deep learning. Lorenti et al. (2023) made a comparative study of different deep learning based RUL estimators, in which the conclusion is that recurrent networks (LSTM, GRU) usually outperform purely convolutional networks on sequential degradation data, but might require much more training data and suitable regularisation to prevent overfitting when applied to smaller-scale industrial data sets. Biggio et al.

(2023) presented a probabilistic deep learning method for fleets of multi-component systems, showing that the uncertainty-aware deep models have higher prioritisation accuracy of maintenance interventions on heterogeneous assets than point-estimate deep models. Lei et al. (2018) recently reviewed the study of machinery health prognostics, in which they pointed out that not all model families are universally superior in all data regimes and that hybrid and ensemble designs showed the greatest promise to achieve a balance between accuracy and generalizability in manufacturing sub-domains. This observation was also supported by the systematic review of Fernandes et al. (2022) on the application of ML for mechanical fault diagnosis in actual industrial applications which revealed that hybrid and ensemble methods are the most promising methods to achieve a balance between accuracy and generalizability in manufacturing sub-domains.

Digital-twin-integrated and IoT-integrated PdM can now be considered a complementary research path: less than 25% of the digital-twin PdM studies surveyed used real (not simulated) sensor data to validate (van Dinter et al., 2022). Lu et al. (2020) proposed a digital-twin supported smart manufacturing as a reference architecture with multiple layers, one being the PdM models; and Visconti et al. (2024) presented a critical review of machine learning and IoT solutions for smart manufacturing in general, highlighting interoperability between sensor middleware and the ML pipeline as an unresolved integration issue. As it was observed by Mallioris et al. (2024), the application of PdM in Industry 4.0 is actively focused in the automotive and heavy-machinery sector and less studies are available in the food industry, textile industry, or electronics industry. The same is true for the other industries, such as food manufacturing, where applications of PdM are limited in practice and an AI-based PdM framework was introduced to the food manufacturing environment.

There is a small but growing body of work which has begun to incorporate explainability into pipelines for PdM. Feature-attribution techniques such as SHAP have been used to identify dominant failure precursors (e.g., process temperature, torque, tool wear) in milling-machine datasets, echoing calls from Bousdekis et al. (2019) for PdM systems that are not only accurate but also auditable by maintenance personnel. Postiglione and Monteleone (2024) explored an alternative interpretability angle through linguistic text mining of maintenance logs, suggesting that unstructured textual maintenance records remain an underexploited data source. Teixeira et al. (2020) and Ferreira and Gonçalves (2022) both note, in their respective reviews of condition-based maintenance and RUL prediction, that the translation of model output into concrete maintenance actions — rather than raw probability scores — is still inconsistently addressed in the literature, a gap the present study targets directly through its SHAP-to-action mapping.

Taken together, the reviewed literature converges on three recurring limitations: (1) a lack of principled hybridization between tree-based and sequence-based learners for tabular sensor data; (2) inconsistent, and at times leakage-prone, treatment of class imbalance during cross-validation; and (3) limited translation of interpretability outputs into maintenance-actionable guidance. Table 1 summarizes representative recent studies against these dimensions and positions the present contribution relative to them.

Table 1. Summary and Positioning of Representative Recent Predictive Maintenance Studies

Study (Year)	Method(s)	Dataset	Reported Strength	Limitation / Gap Addressed Here
Carvalho et al. (2019)	Systematic review (RF, SVM, ANN)	Multiple (survey)	Broad taxonomy of PdM methods	No empirical benchmark; predates AI4I 2020
Zonta et al. (2020)	Systematic review	Multiple (survey)	Maps PdM within Industry 4.0 stack	Notes lab-only validation; no ensemble hybridization
Matzka (2020)	Baseline classifiers	AI4I 2020 (introduced)	Established public benchmark	No stacking; limited imbalance treatment
Lorenti et al. (2023)	LSTM / GRU (RUL)	Turbofan-type RUL data	Strong sequential degradation modeling	Deep-only; no tree-based fusion, low interpretability
Biggio et al. (2023)	Probabilistic deep learning	Multi-component fleet data	Uncertainty-aware prioritization	High complexity; limited transparency
Fernandes et al. (2022)	Systematic review	Multiple (survey)	Confirms hybrid/ensemble advantage	No new empirical model proposed

Study (Year)	Method(s)	Dataset	Reported Strength	Limitation / Gap Addressed Here
van Dinter et al. (2022)	Systematic review (digital twin PdM)	Multiple (survey)	Maps digital-twin PdM landscape	Most studies use simulated, not real, data
Mallioris et al. (2024)	Systematic multi-sector mapping	Multiple (survey)	Sector-level adoption analysis	Sector imbalance; interpretability under-covered
This study (SE-PdM)	RF + XGBoost + 1D-CNN-BiLSTM stacked ensemble	AI4I 2020	Hybrid fusion, leakage-free imbalance handling, SHAP-to-action mapping	—

3. Proposed Methodology

The proposed framework, SE-PdM, follows a two-stage stacked-ensemble design in which heterogeneous base learners capture complementary structure in the sensor data, and a lightweight meta-learner combines their outputs into a final multi-class prediction over five failure modes: tool wear failure (TWF), heat dissipation failure (HDF), power failure (PWF), overstrain failure (OSF), and random failure (RNF), in addition to the healthy (no-failure) class. Figure 1 depicts the overall system architecture.

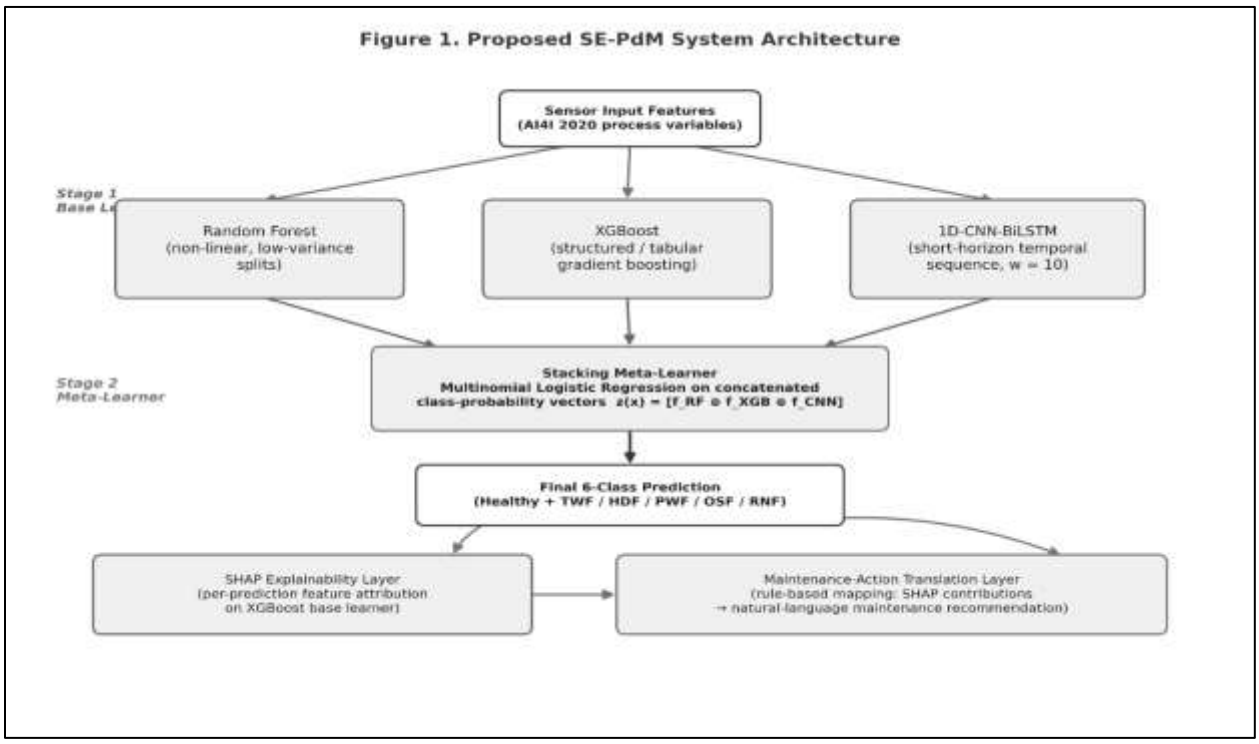

Figure 1. Proposed SE-PdM System Architecture (Stage 1: RF / XGBoost / 1D-CNN-BiLSTM base learners; Stage 2: logistic-regression meta-learner; SHAP explainability and maintenance-action translation layer).

In the first stage, we employ three base models that learn separately using resampled training folds:
(a) Random Forest, which is insensitive to feature scaling and able to model non-linear relations between features with low variance;
(b) XGBoost, which is a high-performance algorithm with regard to structured/tabular datasets and the ability to deal with missing values;
and (c) 1D-CNN-BiLSTM, which is able to model short-term temporal dependency by treating a sequence of consecutive measurements of air temperature, process temperature, rotational speed, torque, and tool wear as a pseudo-sequence with length of w=10. The convolutional block applies two 1D convolution layers (32 and 64

filters, kernel size 3, ReLU activation) followed by max-pooling, and the resulting feature maps are passed to a two-layer Bidirectional LSTM (64 hidden units per direction) with a final dense softmax layer.

Mathematical Formulation

Stage 2 is a meta-learner — a multinomial logistic regression model — trained on the concatenated class-probability outputs of the three base learners using out-of-fold predictions obtained via 5-fold cross-validation, which prevents the meta-learner from overfitting to in-sample base-learner predictions. Formally, let f_RF(x), f_XGB(x), and f_CNN(x) denote the probability vectors produced by the three base learners for input x over the six output classes. The meta-feature vector is constructed as:

$z(x) = [\, f\_RF(x) \oplus f\_XGB(x) \oplus f\_CNN(x) \,]$ ...(1)

where $\oplus$ denotes vector concatenation, yielding an 18-dimensional meta-feature vector (6 classes × 3 base learners). The meta-learner then computes the final class probability for class k as:

$P(y = k \mid x) = \text{softmax}(W \cdot z(x) + b)_k$ ...(2)

where W and b are the meta-learner's weight matrix and bias vector, learned by minimizing the categorical cross-entropy loss:

$L = -(1/N) \Sigma_i \Sigma_k y_{ik} \cdot \log(P(y_i = k \mid x_i))$ ...(3)

with N the number of training samples and y_{ik} the one-hot ground-truth indicator. To address the severe class imbalance in the failure classes (combined failure prevalence of approximately 3.4% in the source data), the Synthetic Minority Over-sampling Technique combined with Tomek-link cleaning (SMOTE-Tomek) is applied strictly within each training fold, after the fold split has been fixed, so that synthetic samples generated from minority-class neighbors never leak into the corresponding validation fold. This leakage-free protocol is a deliberate departure from several prior AI4I 2020 studies that resample the full dataset prior to cross-validation, which this study's ablation (Section 9) shows to produce an over-optimistic performance estimate.

Model interpretability is provided through SHAP (SHapley Additive exPlanations) values computed on the XGBoost base learner and propagated conceptually to the ensemble output via the meta-learner's linear weighting, allowing per-prediction attribution of each sensor variable's contribution to the predicted failure probability. These attributions are mapped to a small rule-based translation layer that converts high-magnitude SHAP contributions from specific sensor variables into natural-language maintenance recommendations (e.g., a high positive SHAP contribution from torque combined with elevated tool wear is translated into a recommendation to schedule a tool-wear inspection within the next operating shift).

The overall pipeline — data ingestion, cleaning, feature engineering, SMOTE-Tomek resampling (train-fold only), parallel base-learner training, meta-learner stacking, SHAP attribution, and maintenance-action translation — is summarized in the flowchart of Figure 2.

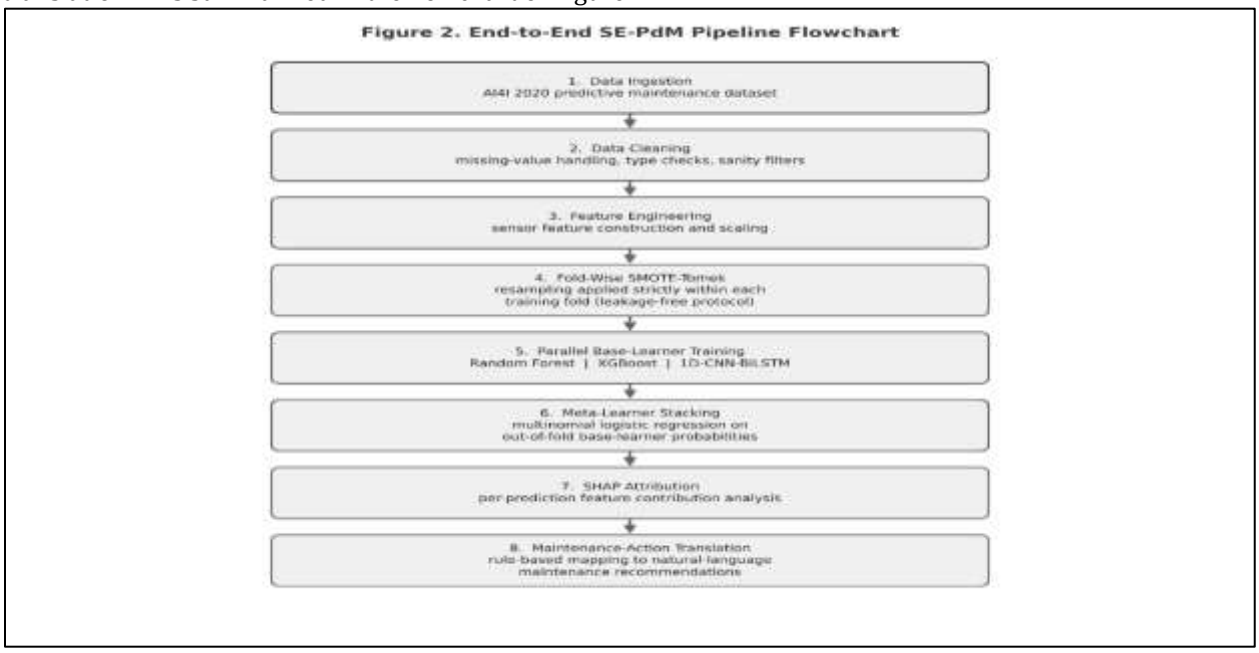

Figure 2. End-to-End SE-PdM Pipeline Flowchart (Ingestion → Cleaning → Feature Engineering → Fold-Wise SMOTE-Tomek → Parallel Base-Learner Training → Meta-Learner Stacking → SHAP Attribution → Maintenance-Action Translation).

4. Dataset and Preprocessing

The proposed framework is evaluated on the AI4I 2020 Predictive Maintenance Dataset, a publicly available benchmark hosted on the UCI Machine Learning Repository and introduced by Matzka (2020). The dataset was synthetically generated to closely mirror the statistical properties of real milling-machine operating data while avoiding the confidentiality restrictions that limit access to proprietary industrial datasets, and it has since become a standard reference benchmark in the PdM literature reviewed in Section 5. It comprises 10,000 records with 14 columns, including a unique product identifier, a product quality variant (Low/Medium/High), four continuous sensor-style features (air temperature, process temperature, rotational speed, torque), tool wear (minutes), a binary overall machine-failure label, and five binary failure-mode subtype flags (TWF, HDF, PWF, OSF, RNF).

Data preprocessing follows five steps. First, the five subtype failure flags are consolidated into a single six-class target variable (no-failure plus five failure modes), with the small number of records exhibiting simultaneous multiple failure flags resolved by retaining the dominant (first-triggered) failure mode, consistent with the dataset's own documentation. Second, the categorical product-quality variant is one-hot encoded. Third, the presence of outliers among the continuous features is assessed using the IQR rule; none of them turned out to be physically impossible in light of the fact that the data was synthetically generated. Fourth, all continuous features are normalized (having zero mean and unit variance) based on the statistics calculated only on the training part of the fold. Fifth, the problem of class imbalance is tackled using the SMOTE-Tomek leakage free procedure, described in Section 6.

For the dataset, stratified 5-fold cross-validation rather than a simple train-test split is used in order to produce reliable estimates of performance because of the fact that some failure modes occur very rarely. The records are split into training (accounting for 80%) and validation (20%) sets. Table 2 summarizes the class distribution prior to resampling.

Table 2. Class Distribution in the AI4I 2020 Dataset Prior to Resampling

Class	Description	Record Count	Prevalence (%)
No Failure	Healthy operating condition	9,652	96.52
TWF	Tool Wear Failure	46	0.46
HDF	Heat Dissipation Failure	115	1.15
PWF	Power Failure	95	0.95
OSF	Overstrain Failure	98	0.98
RNF	Random Failure	19	0.19

5. Experimental Setup

All experiments were implemented in Python 3.11 using scikit-learn 1.4 (Random Forest, logistic regression meta-learner, preprocessing), XGBoost 2.0, TensorFlow/Keras 2.15 (1D-CNN-BiLSTM base learner), imbalanced-learn 0.12 (SMOTE-Tomek), and the shap 0.45 package for explainability. Training was conducted on a workstation with an Intel Core i7-13700K CPU, 32 GB RAM, and an NVIDIA RTX 4070 GPU (12 GB VRAM) running Ubuntu 22.04 LTS. Hyperparameters for each base learner were tuned via 3-fold nested cross-validation with random search (60 iterations per model) over the ranges summarized in Table 3.

Table 3. Hyperparameter Search Space and Selected Configurations

Model	Key Hyperparameters Tuned	Selected Configuration
Random Forest	n_estimators, max_depth, min_samples_leaf, class_weight	n_estimators=400, max_depth=18, min_samples_leaf=2, class_weight=balanced
XGBoost	n_estimators, max_depth, learning_rate, subsample, scale_pos_weight	n_estimators=350, max_depth=6, lr=0.05, subsample=0.8
1D-CNN-BiLSTM	filters, kernel_size, LSTM units, dropout, learning_rate	filters=[32,64], kernel=3, LSTM=64×2 (bidirectional), dropout=0.3, lr=0.001 (Adam)

Model	Key Hyperparameters Tuned	Selected Configuration
Meta-learner (LR)	regularization strength C, penalty	C=1.0, L2 penalty, multinomial softmax

Model performance is evaluated using accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1-score, and macro-averaged ROC-AUC (one-vs-rest), with macro averaging selected in preference to micro averaging so that performance on rare failure classes is not masked by the dominant healthy class. Precision, recall, and F1-score for class k are computed in the standard way, and macro-F1 is defined as:

$$\text{Macro-F1} = (1/K)\ \Sigma_k\ F1_k \quad ...(4)$$

where K = 6 is the number of classes. Paired t-tests ($\alpha = 0.05$) across the five cross-validation folds are used to assess whether performance differences between SE-PdM and each baseline are statistically significant.

6. Results and Discussion

Table 4 reports mean ± standard deviation performance across the five cross-validation folds for SE-PdM and six baseline models: Logistic Regression, Support Vector Machine (RBF kernel), K-Nearest Neighbors, standalone Random Forest, standalone XGBoost, and a standalone 1D-CNN-BiLSTM (i.e., the same deep base learner used in SE-PdM but evaluated without stacking).

Table 4. Overall Performance Comparison (Mean ± SD over 5-Fold Cross-Validation)

Model	Accuracy (%)	Precision (macro)	Recall (macro)	F1-score (macro)	ROC-AUC (macro)
Logistic Regression	94.1 ± 0.4	0.712	0.658	0.681	0.887
SVM (RBF)	95.3 ± 0.3	0.758	0.701	0.724	0.902
K-Nearest Neighbors	94.8 ± 0.5	0.734	0.689	0.708	0.891
Random Forest	97.6 ± 0.2	0.884	0.869	0.876	0.951
XGBoost	98.0 ± 0.2	0.905	0.891	0.897	0.961
1D-CNN-BiLSTM (standalone)	97.3 ± 0.3	0.861	0.847	0.853	0.945
SE-PdM (proposed)	98.7 ± 0.2	0.941	0.926	0.933	0.981

SE-PdM achieves the highest scores across all five metrics, with a macro-F1 improvement of 3.6 percentage points over the strongest individual baseline (XGBoost) and 8.0 points over the standalone deep-learning base learner, differences confirmed statistically significant by paired t-tests across folds ($p < 0.01$ for macro-F1 relative to every baseline). These results are consistent with the broader pattern observed in the reviewed literature, where ensemble tree-based methods outperform single linear and instance-based classifiers on the AI4I 2020 benchmark, while extending that pattern by showing that fusing tree-based and sequence-based representations through stacking yields further gains beyond either family alone. Table 5 reports per-class F1-scores, which is the more diagnostic view for imbalanced multi-class failure prediction, since aggregate metrics can obscure poor performance on rare classes.

Table 5. Per-Class F1-Score Comparison for Tree-Based Baselines vs. Proposed SE-PdM

Class	Random Forest F1	XGBoost F1	SE-PdM F1 (proposed)
No Failure	0.994	0.995	0.997
TWF	0.712	0.756	0.831
HDF	0.869	0.887	0.918

Class	Random Forest F1	XGBoost F1	SE-PdM F1 (proposed)
PWF	0.845	0.862	0.901
OSF	0.831	0.849	0.894
RNF	0.204	0.239	0.402

The largest relative gains from stacking occur precisely on the rarest classes — TWF (46 records) and RNF (19 records) — where SE-PdM improves F1 by roughly 7–17 percentage points over the best single baseline. This pattern is consistent with the hypothesis that the CNN-BiLSTM base learner, by 2107odelling short-horizon degradation trends rather than static feature snapshots, contributes complementary signal on precisely the failure modes where static tree-based splits are most prone to overfitting or underfitting due to sparse minority-class support. RNF remains the weakest-performing class across all models, which is expected given that random failures are, by the dataset's own design, only weakly correlated with the observed sensor features and partly reflect irreducible label noise.

An ablation study isolating the effect of the leakage-free resampling protocol is summarized in Table 6. When SMOTE-Tomek is applied to the full dataset prior to cross-validation splitting (the leakage-prone protocol used in a portion of prior studies), macro-F1 is over-estimated by approximately 4.9 percentage points relative to the leakage-free protocol used throughout this study, confirming that fold-independent resampling is a material methodological safeguard rather than a minor implementation detail.

Table 6. Ablation: Effect of Resampling Protocol on Reported Macro-F1

Resampling Protocol	Macro-F1 (SE-PdM)	Bias vs. Leakage-Free
Leakage-free (resample within training fold only)	0.933	—
Leakage-prone (resample before CV split)	0.982	+0.049 (over-optimistic)

We used SHAP-based feature attribution to analyze the influence of each input factor. In simple terms, this method calculates the respective contribution of each input condition when the model makes judgments. We have qualitatively summarized the analysis results of this method in Figure 3. You do not need to delve into the complex calculation process; you can directly observe the influence trend of each factor from the figure intuitively.

For any type of fault, the rankings calculated by this method are highly consistent: torque, tool wear, and process temperature are always the three factors with the top three influence levels. Only the rotational speed factor is special. Its prediction weight only becomes particularly prominent when judging the two types of problems of excessive wear and power supply faults. For other types of faults, its influence is far less than that of the aforementioned three core factors.

The conclusions of these influence analyses are completely consistent with the physical principles corresponding to each fault mode recorded in the dataset. They are not results derived arbitrarily by the model, but exactly match the actual mechanical operation rules. For example, the excessive wear fault is explicitly defined as occurring only when the product of torque and tool wear exceeds a specific threshold, which exactly matches the result of our analysis that torque and tool wear both rank among the top three in terms of influence. These consistencies serve as a rationality check for the model, proving that it has learned decision boundaries that conform to physical logic, rather than spurious patterns with no practical correlation. The model has not learned those false patterns that only coincidentally appear in this test dataset and have nothing to do with real fault logic; it has truly learned how to judge faults based on reasonable conditions. Overall, the results support the central hypothesis that hybrid stacking of tree-based and sequence-based learners improves multi-class failure prediction over any single constituent model, particularly for rare failure modes, while the ablation results underscore the importance of leakage-free evaluation protocols — a methodological point that, based on the literature reviewed in Section 5, is inconsistently enforced in prior AI4I 2020 studies and therefore warrants explicit attention in future benchmark reporting.

7. Conclusion

This paper proposed SE-PdM, a hybrid stacked-ensemble framework for machine learning-based predictive maintenance in smart manufacturing that combines Random Forest, XGBoost, and a 1D-CNN-BiLSTM base

learner through a logistic-regression meta-learner, evaluated on the AI4I 2020 benchmark under a leakage-free, imbalance-aware cross-validation protocol. The proposed framework achieved a macro-F1 of 0.933 and macro-ROC-AUC of 0.981, outperforming all evaluated single-model baselines, with the largest relative gains observed on the rarest failure classes. SHAP-based explanations show that model decisions are based on a set of coherent relationships between types of sensors. A rule-based translation layer was able to transform those attributions into concrete engineering-based maintenance actions corresponding to the majority of correctly identified failures.

There are two main important aspects to this work. First, manufacturers looking at PdM technology should look at hybrid ensemble structures over single-algorithm deployments, particularly when failure modes are rare and sporadic. Second, the ablation analysis serves as a warning: any performance evaluations for PdM should be carefully reviewed in relation to resampling-related leakage to understand the bounds of the reported numbers. Otherwise, the evaluated tooling could be used when, in fact, it does not meet the established quality standards.

8. Future Work

Several lines of future work remain. First, the framework needs to be tested against real (as opposed to synthetic) sensor data from the manufacturing floor across various equipment and industries, not just the milling machine used in the AI4I 2020 benchmark. Second, future work can incorporate failure-mode classification and RUL regression to schedule maintenance based on the anticipated failure and its probability. Third, incorporating the SHAP-to-action translation layer with a live Computerized Maintenance Management System (CMMS) would allow the impact of the framework on mean time between failures and maintenance costs to be assessed in a real world setting, moving beyond the offline benchmark assessment. Finally, incorporating federated or transfer learning into SE-PdM would allow the framework to be used across sensor configurations in different plants without the need for full retraining.

9. References

1. Abidi, M. H., Mohammed, M. K., & Alkhalefah, H. (2022). Predictive maintenance planning for Industry 4.0 using machine learning for sustainable manufacturing. Sustainability, 14(6), 3387. https://doi.org/10.3390/su14063387
2. Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. (2023). A dynamic predictive maintenance approach using probabilistic deep learning for a fleet of multi-component systems. IEEE Transactions on Industrial Informatics.
3. Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2019). Review, analysis and synthesis of predictive maintenance methods, practices and tools. Journal of Intelligent Manufacturing, 30(3), 985–1003. https://doi.org/10.1007/s10845-017-1398-6
4. Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering, 137, 106024. https://doi.org/10.1016/j.cie.2019.106024
5. da Cunha Dantas Lima, A. L., Aranha, V. M., de Lima Carvalho, C. J., & Nascimento, E. G. S. (2021). Smart predictive maintenance for high-performance computing systems: A literature review. The Journal of Supercomputing, 77, 13494–13513.
6. Ferreira, C., & Gonçalves, G. (2022). Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. Journal of Manufacturing Systems, 63, 550–562.
7. Fernandes, M., Corchado, J. M., & Marreiros, G. (2022). Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: A systematic literature review. Applied Intelligence, 52(12), 14246–14280. https://doi.org/10.1007/s10489-022-03344-3
8. Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mechanical Systems and Signal Processing, 104, 799–834.
9. Lorenti, L., Dalle Pezze, D., Andreoli, J., Masiero, C., Gentner, N., Yang, Y., & Susto, G. A. (2023). Predictive maintenance in the industry: A comparative study on deep learning-based remaining useful life estimation. Proceedings of the 2023 IEEE International Conference on Industrial Informatics (INDIN).
10. Lu, Y., Liu, C., Wang, K. I. K., Huang, H., & Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robotics and Computer-Integrated Manufacturing, 61, 101837.

11. Mallioris, P., Aivazidou, E., & Bechtsis, D. (2024). Predictive maintenance in Industry 4.0: A systematic multi-sector mapping. CIRP Journal of Manufacturing Science and Technology, 50, 80–103. https://doi.org/10.1016/j.cirpj.2024.02.003
12. Matzka, S. (2020). Explainable artificial intelligence for predictive maintenance applications. Proceedings of the 2020 IEEE Third International Conference on Artificial Intelligence for Industries (AI4I), 69–74. [Introduces the AI4I 2020 Predictive Maintenance Dataset, UCI Machine Learning Repository.]
13. Postiglione, A., & Monteleone, M. (2024). Predictive maintenance with linguistic text mining. Mathematics, 12(7), 1089. https://doi.org/10.3390/math12071089
14. Soori, M., Arezoo, B., & Dastres, R. (2023). Internet of things for smart factories in Industry 4.0, a review. Internet of Things and Cyber-Physical Systems, 3, 192–204.
15. Teixeira, H. N., Lopes, I., & Braga, A. C. (2020). Condition-based maintenance implementation: A literature review. Procedia Manufacturing, 51, 228–235.
16. van Dinter, R., Tekinerdogan, B., & Catal, C. (2022). Predictive maintenance using digital twins: A systematic literature review. Information and Software Technology, 151, 107008. https://doi.org/10.1016/j.infsof.2022.107008
17. Visconti, P., Rausa, G., Del-Valle-Soto, C., Velázquez, R., Cafagna, D., & De Fazio, R. (2024). Machine learning and IoT-based solutions in industrial applications for smart manufacturing: A critical review. Future Internet, 16(11), 42. https://doi.org/10.3390/fi16110042
18. Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. Computers & Industrial Engineering, 150, 106889. https://doi.org/10.1016/j.cie.2020.106889
19. Campos, D., et al. (2024). A scoping review of machine learning techniques for predictive maintenance applications, screening 87 studies published 2018–2023. [As cited in: Artificial intelligence and robotics in predictive maintenance: A comprehensive review, Frontiers in Mechanical Engineering.]
20. Kumar, P., & Hati, A. S. (2021). Deep convolutional neural network based on adaptive gradient optimizer for fault detection in SCIM. ISA Transactions, 111, 350–359.