



Baština Akademije nauka i umjetnosti Bosne i Hercegovine

SMART MANUFACTURING ON THE HORIZON: A Conceptual Shift from Industry 4.0 to Industry 5.0: Industrija

Karabegović, Isak

2026

Academy of Sciences and Arts of Bosnia and Herzegovina

<https://bastina.anubih.ba/handle/123456789/875>

Preuzeto s Baštine Akademije nauka i umjetnosti Bosne i Hercegovine

<https://bastina.anubih.ba/>

An Explainable Artificial Intelligence Framework for Predictive Maintenance Using Hybrid Machine Learning and Multi-Task Transformer Models

Milica Tufegdžić^{*1}

Abstract: *An integrated explainable artificial intelligence (XAI) framework for predictive maintenance is proposed, focusing on failure prediction, Remaining Useful Life (RUL) estimation, and real-time asset monitoring. The framework is evaluated using a synthetically generated predictive maintenance data simulating 50 machines over time, producing realistic multivariate sensor signals that degrade with wear and occasional shocks, then converted into a supervised dataset for failure prediction and RUL estimation. The study evaluates several classical machine learning (ML) models, such as XGBoost (Extreme Gradient Boosting), Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LogR), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), for classification and regression tasks. XGBoost demonstrated strong overall performance, achieving the best RUL regression performance and competitive classification results, while RF and GB achieved comparable or slightly better results for individual classification metrics. A multi-task transformer is introduced to jointly model failure probability, uncertainty-aware RUL, anomaly reconstruction, and survival-based time-to-failure estimation, thereby providing a unified framework for modeling multiple temporal predictive-maintenance objectives. An autoencoder is included as a supplementary reconstruction-based component for anomaly scoring, while a risk function combines failure probability and RUL to enable real-time decision-making. To ensure interpretability, SHapley Additive exPlanations (SHAP) analysis is used to quantify feature importance at the sensor level. The framework is evaluated using Receiver Operating Characteristic (ROC) curves and Precision-Recall curves, along with Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2) metrics. These results demonstrate the strong performance of ensemble models, particularly XGBoost. Overall, the framework demonstrates the potential of combining tree-based models, transformer-based temporal learning, and SHAP-based interpretability for predictive maintenance under synthetic sensor conditions.*

Keywords: multi-task learning, transformer models, failure prediction, remaining lifetime estimation, anomaly detection, industrial monitoring

^{*1}Academy of Professional Studies Šumadija, Kragujevac, Serbia
E-mail: mtufegdzc@asss.edu.rs

1. Introduction

In modern industrial environments, the increasing reliance on interconnected cyber-physical systems has intensified the need for intelligent tools that ensure operational reliability and cost efficiency. Predictive Maintenance (PdM) has emerged as a transformative approach that leverages data-driven methods to anticipate equipment failures and optimize maintenance schedules, thereby minimizing unexpected downtime and extending asset life cycles. However, despite significant progress in machine learning (ML)-based PdM solutions, challenges remain in handling high-dimensional temporal sensor data, capturing complex degradation behaviors, and providing transparent, interpretable insights into model decisions. These challenges hinder the adoption of AI-driven maintenance systems in safety-critical and regulated industrial applications.

Predictive maintenance is a condition-based approach that uses machine condition data to detect and predict faults before equipment fails. Diagnostic focuses on detecting and identifying faults using either hardware redundancy (multiple sensors) or analytical redundancy based on models and data. Prognostics goes a step further by predicting future failures and estimating the RUL of equipment. Today, data-driven ML methods are widely used in predictive maintenance because Internet of Things (IoT) systems generate large amounts of industrial data, making these techniques effective and cost-efficient for complex manufacturing systems [1]. Through preprocessing and analytics, this data can be transformed into meaningful insights about manufacturing processes, supporting decision-making, maintenance scheduling, failure detection, Remaining Useful Life (RUL) and health index estimation, cost reduction, and improved worker safety [2].

PdM uses historical data, real-time sensor readings, and ML algorithms to predict equipment failures before they occur. It analyzes patterns and detects anomalies to reduce unplanned downtime, optimize maintenance schedules, lower costs, and extend asset lifespan [3]. PdM leverages real-time data analytics and ML-based models for efficient maintenance scheduling and improved equipment longevity [4]. The aim of PdM is not only to collect process parameters, but also to monitor the physical health of equipment using data such as pressure, vibration, temperature, acoustics, and flow rate. This information is used for fault detection, early anomaly identification, equipment health assessment, and prediction of future system states [5].

Predictive maintenance is further advanced by the emergence of the Artificial Intelligence of Things, which integrates IoT sensor networks with Artificial Intelligence (AI) algorithms to analyze real-time operational data for anomaly detection, fault prediction, and maintenance recommendations. This integration enables autonomous, scalable, and intelligent maintenance systems aligned with Industry 5.0, while edge computing, federated learning, and hybrid AI models

enhance responsiveness, privacy, and adaptability [6]. RUL prediction is a core aspect of predictive maintenance that reduces downtime and maintenance costs while estimating the remaining operational life of a system or component before failure [7].

Recent advancements in Explainable Artificial Intelligence (XAI) offer a promising pathway toward trustworthy and interpretable PdM systems. By integrating XAI techniques, models can not only predict failures and estimate RUL more accurately but also justify their predictions through meaningful, human-understandable explanations [8]. This is particularly crucial in industrial contexts where maintenance decisions must be justified to engineers, operators, and decision-makers. XAI in predictive maintenance includes various methods grouped into model-agnostic, model-specific, and hybrid approaches [9]. Hybrid models that integrate XAI with traditional ML are used in manufacturing, predictive maintenance, and quality control to improve decision-making and support human-AI collaboration [10]. Moreover, combining interpretable ensemble learning models with temporal deep learning architectures enables more robust generalization across varying operational conditions and sensor noise levels [11].

This study proposes an integrated XAI Framework for Predictive Maintenance, combining hybrid ML models and a multi-task transformer architecture. The framework jointly addresses failure prediction, uncertainty-aware RUL estimation, anomaly detection, and time-to-failure modeling, while ensuring interpretability at the feature and sensor levels. Synthetic multivariate time-series data simulating machinery degradation and shock events are used to evaluate model performance under realistic conditions. Classical ML models, such as Extreme Gradient Boosting (XGBoost), Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LogR), Linear regression (LR), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), are benchmarked to select optimal classifiers and regressors for PdM tasks. Ensemble methods, particularly XGBoost, demonstrate strong performance across the evaluated classification and regression tasks.

To enhance temporal modeling and uncertainty quantification, a multi-task transformer is introduced, complemented by an autoencoder as a supplementary reconstruction-based component for anomaly scoring. A composite risk function integrating failure probability and RUL is proposed to support real-time maintenance decision-making. Additionally, SHapley Additive exPlanations (SHAP) are employed to quantify sensor-level contributions, promoting transparency and operational trust. Evaluations using Receiver Operating Characteristic (ROC) and Precision-Recall curves, along with Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2) metrics, confirm the effectiveness of the proposed hybrid and explainable approach.

Overall, this framework bridges the gap between predictive performance and interpretability, demonstrating that the integration of ensemble tree models, transformer-based temporal learning, and SHAP-driven explainability can significantly enhance the robustness, accuracy, and transparency of predictive maintenance systems-laying the groundwork for next-generation intelligent and trustworthy industrial asset management.

2. Literature Review

The AI predictive maintenance framework that combines Convolutional Neural Network (CNN) for spatial feature extraction, Gated Recurrent Unit (GRU) for temporal pattern learning, and Transformers for long-range dependencies, using ensemble stacking and Bayesian Optimization for stable and efficient model tuning, was evaluated on real smart manufacturing multi-sensor data, including vibration, temperature, acoustic, and pressure signals, to monitor machine condition over time [12]. RF, XGBoost, SVM, and ANN models were evaluated using standard performance metrics. Results show that RF and XGBoost achieved the best accuracy and robustness, demonstrating the value of data-driven fault prediction for reducing downtime and improving proactive maintenance [13].

The approach, which combines a Transformer backbone with Monte Carlo Dropout and Conformal Prediction to capture epistemic and aleatoric uncertainty while improving calibration through Bayesian approximation and distribution-free intervals, enabling reliable confidence estimates that support early fault detection and prioritized maintenance, was validated on the NASA C-MAPSS dataset and benchmarked against existing methods [14]. Light Gradient Boosting Machine (LightGBM) model with Dropouts meet Multiple Additive Regression Trees booster tuning was developed for predictive maintenance using five datasets from 100 machines containing telemetry, error, maintenance, and failure data. The data were processed and feature-engineered, and the trained model was used to classify short-term failure risk [15]. A conceptual NLP-Hybrid Intelligence framework is proposed for smart manufacturing condition monitoring and predictive maintenance, integrating sensor analytics with NLP-based contextual industrial knowledge [16]. A hybrid predictive maintenance framework that combines Long Short-Term Memory (LSTM) autoencoder-based unsupervised anomaly detection with supervised fault prediction using an MLP was validated on a digital-twin dataset simulating five years of industrial operation and addresses class imbalance and operational variability [11].

ML algorithms used for RUL prediction in milling operations include Stochastic Gradient Descent (SGD) Regressor, RF Regressor, Decision Tree Regressor, Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP) [17]. RUL prediction models were developed using LR, SVR, RF, XGBoost, ridge regression, and artificial neural networks (ANN). This ML-based RUL prediction

was combined with Weibull distribution-based reliability analysis and a block replacement model with minimal repair to optimize preventive maintenance strategies [18]. ML algorithms such as SVR, RF, and XGBoost are used to predict the RUL of lithium-ion batteries based on real-world degradation data [19]. The model, which applied normalization and PCA for preprocessing, used interpolation and grid search for optimization, and employed MLP to predict engine RUL, was tested on NASA turbo engine datasets [20]. PdM framework combines a probabilistic neural network for RUL prediction with a Quantile regression deep Q-network distributional reinforcement learning agent. The model analyzes sensor data to estimate RUL, measure prediction uncertainty, and assess failure probability [21].

A data-driven ML approach using ANN with Neural Basis Networks combined with Fully Convolutional Networks and Autoencoder-based Convolutional Networks was developed to model degradation paths, predict failures, and estimate RUL in complex manufacturing systems. The method improves prediction accuracy by using an enhanced NBN training algorithm with an accumulative design that preserves the monotonic degradation trend and effectively handles large, complex multi-machine datasets [22].

A hybrid approach combining supervised learning using a MLP and reinforcement learning using Q-learning was proposed to improve RUL prediction accuracy and precision, and was evaluated on the C-MAPSS dataset [23]. A hybrid predictive maintenance framework combined physics-based modeling with a bidirectional Long Short-Term Memory (BiLSTM)-based AI model, using model residuals to enhance fault diagnosis, anomaly detection, and RUL prediction for rotating machinery such as bearing wear and shaft imbalance [24]. The hybrid deep learning model that was developed for RUL point predictions, consisted of an input layer defined by the window size and number of features, followed by multiple 1D-CNN layers with ReLU activation for feature extraction and Bi-LSTM layers for temporal learning. Fully connected dense layers with Exponential Linear Unit activation further process the learned representations, and a final linear output layer with a single neuron generates the prediction [25].

A RUL prediction method that combines BiLSTM and Transformer architectures is proposed. First, a BiLSTM-based denoising autoencoder extracts features from multivariate sensor data, which are then used by a Transformer encoder to estimate RUL [26]. A multi-task transformer encoder and customized gate control model was proposed to jointly predict tool RUL and wear stages using a transformer encoder for shared feature extraction and customized gate control for task-specific and shared features. It was trained using a dynamic adaptive multi-task loss function to improve learning efficiency [27]. The Multi-Input Data-1DC+LRT model was proposed as a multi-task learning framework integrating a multi-input 1D CNN (and a low-rank transformer to process a high-dimensional

sensor, multi-condition, and health indicator data while reducing computational complexity, in order to predict RUL [28].

XAI improves AI-driven maintenance by making ML models more transparent, interpretable, and trustworthy, addressing the "black box" nature of many systems and enabling smarter decision-making. However, challenges remain in integrating XAI with IoT-based digital twins, balancing explainability with cybersecurity, and ensuring scalability for reliable and efficient deployment [29]. A XAI framework for failure detection and classification uses symbolic expressions from a genetic programming classifier. The method applies normalization, oversampling, cross-validation, and hyperparameter tuning, then builds a meta-dataset to train RF models and form class-specific voting ensembles [30]. A hybrid ensemble predictive maintenance framework combines multiple ML classifiers to overcome key limitations in PdM, integrating RF, XGBoost, and LightGBM with SHAP for transparent and interpretable decision-making. The models are trained using 5-fold cross-validation, with soft-voting and weighted probability averaging for final prediction, while hyperparameter tuning and SHAP analysis are applied within a unified pipeline to ensure reproducibility and explainability [31].

RUL estimation on the NASA C-MAPSS dataset is performed using a stacked LSTM model for temporal degradation learning. SHAP and TimeSHAP provide sensor- and time-step-level interpretability, while robustness under Gaussian, drift, and spike noise is evaluated using Spearman correlation and Top-K consistency metrics [32]. A predictive maintenance framework for aircraft engines was developed that combines accurate RUL prediction with cost-aware maintenance scheduling using an ensemble of LightGBM, CatBoost, and GB to improve prediction accuracy and stability, while SHAP analysis enhances interpretability by identifying key sensors and operational factors [33].

3. Methodology

3.1. Synthetic Dataset Generation for Predictive Maintenance

A fully synthetic dataset was constructed to address the limited availability of real industrial condition monitoring data, which is typically constrained by failure scarcity, class imbalance, and high acquisition cost. This simulation-based setup provides a controlled environment for modeling machine degradation dynamics and evaluating Transformer-based predictive maintenance architectures under consistent conditions.

The dataset represents a fleet of machines operating under heterogeneous conditions over time. Its structure is defined through configurable parameters, including the number of machines, temporal resolution per unit, and sliding-window length. Each machine trajectory is generated independently to reflect

variability across industrial assets, matching the multi-sequence input formulation used in the Transformer model.

Machine health evolution is modeled through a nonlinear wear process combined with stochastic shock events that emulate abrupt stress, overload, and operational disturbances. These mechanisms generate multivariate operational signals, including temperature, vibration, pressure, current, and wear-related measurements. The raw sensor observations are subsequently transformed into engineered features for the real-time inference pipeline. Specifically, statistical descriptors and condition indicators are calculated from the sensor measurements, including mean, standard deviation, maximum value, the latest wear state, and an aggregated vibration/shock measure. The resulting feature representation is described explicitly in Section 3.2. Degradation manifests as increasing temperature and vibration trends and decreasing pressure, while Gaussian noise is introduced to emulate measurement uncertainty and environmental variability. Failure generation follows a stress-driven probabilistic mechanism in which accumulated degradation increases the likelihood of failure over time. A stochastic decision rule determines event occurrence at each time step, and failures are assigned to predefined classes such as thermal, vibration-related, and pressure leakage faults. This formulation supports both binary failure prediction and more granular diagnostic learning consistent with multi-task outputs.

For real-time inference, incoming sensor observations are transformed into a fixed feature representation using a dedicated feature-engineering procedure. The procedure calculates statistical descriptors of temperature, vibration, and pressure, together with the mean current, the most recent wear value, and an aggregated vibration-based shock indicator. A machine identifier is retained separately for machine-level traceability. This engineered representation is subsequently passed to the trained predictive models during streaming inference. Supervision signals are generated in parallel with the input construction. Binary labels indicate whether failure occurs within a future prediction horizon, while Remaining Useful Life (RUL) quantifies time-to-failure. The resulting dataset is stored in CSV format and contains engineered features, machine identifiers, and target variables, directly supporting the multi-task learning framework of failure prediction, RUL estimation, and related auxiliary objectives used in the Transformer architecture.

3.2. Dataset Analysis

The dataset comprises 1,350 observations collected from 50 distinct machines. Two variables, label and RUL, are used as target variables. The label variable represents binary failure status, whereas RUL represents the remaining useful life in cycles. The dataset contains 13 columns in total, comprising ten engineered operational features, one machine identifier (`machine_id`), and two target variables (label and RUL).

The conventional machine-learning benchmark, streaming-oriented predictive analysis, and SHAP-based interpretability analysis use a common engineered predictor representation. The feature-engineering procedure generates ten operational features: temp_mean, temp_std, temp_max, vib_mean, vib_std, pressure_mean, pressure_std, current_mean, wear_last, and shock_sum. These engineered features constitute the input representation used by XGBoost, Random Forest, Gradient Boosting, Logistic Regression, SVM, KNN, and their regression counterparts, ensuring a consistent feature space across the conventional machine-learning benchmark. The same engineered feature representation is also used in the real-time streaming procedure and for SHAP analysis of the XGBoost classification model. The machine_id variable is retained solely as an identifier and is excluded from the input features of all predictive models. It is used only for machine-level identification, traceability, and data partitioning where applicable. The distinction between the different variable categories can therefore be summarized as follows: (i) raw/original predictor variables are the variables present in the dataset after the exclusion of the two targets; (ii) engineered features are the ten derived sensors and condition descriptors generated by the streaming feature-engineering procedure; (iii) machine_id is an identifier used for machine-level traceability; and (iv) label and RUL are target variables used for classification and regression, respectively. In addition to these sensor measurements, the dataset contains two target variables: a binary failure label indicating normal versus failure/alert conditions, and a continuous RUL value ranging from 0 to 270 cycles.

The dataset exhibits a pronounced class imbalance in the failure label, with 1,145 samples (84.8%) corresponding to normal operation and 205 samples (15.2%) representing failure states. This imbalance reflects real industrial settings where failures are relatively rare compared to normal operating conditions and must be accounted for during model development to avoid biased predictions. The boxplot analysis (Figure 1) further confirms clear separation between normal and failure states across key operational metrics. Failure conditions are associated with higher temperature (≈ 85.4 °C vs. 78.4 °C), increased vibration (2.67 mm/s vs. 1.67 mm/s), elevated tool wear (median ≈ 0.71 vs. 0.32), and reduced pressure (≈ 28.8 bars vs. 31.6 bars). These shifts are consistent and non-overlapping in several cases, particularly for vibration and wear, indicating strong discriminative power for classification tasks.

From a modeling perspective, these patterns suggest high separability between classes, making the dataset suitable for supervised learning approaches such as RF or XGBoost. The monotonic trends across features also indicate potential utility for early anomaly detection and predictive maintenance systems.

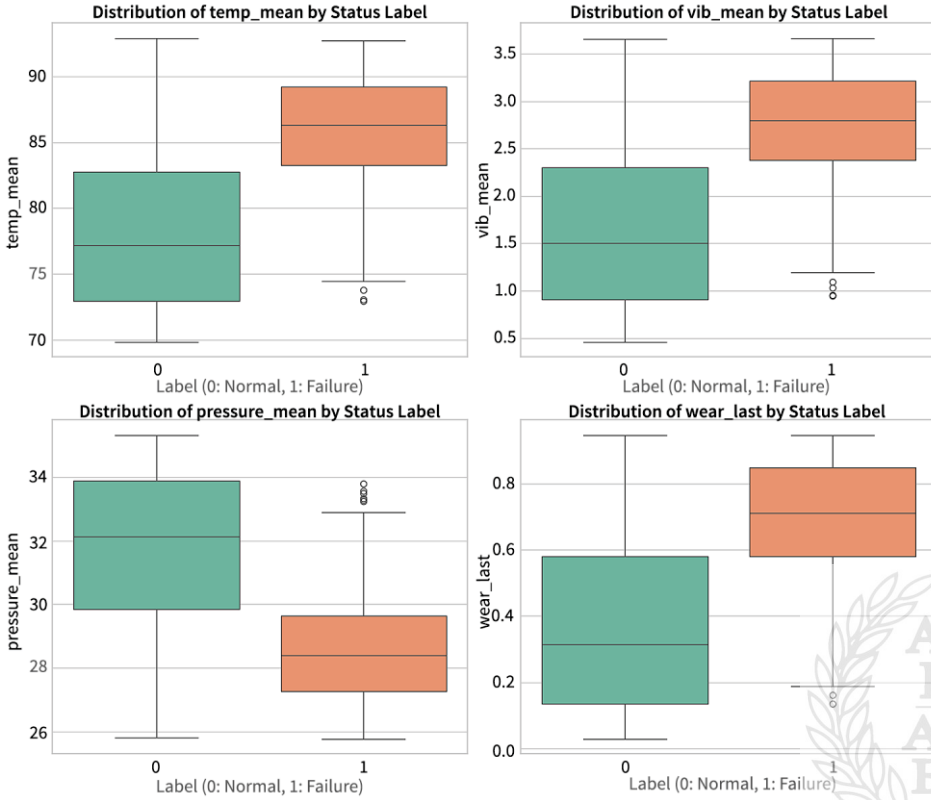


Figure 1. Boxplot analysis

The RUL distribution (Figure 2) is heavily right-skewed, with most values concentrated between 0 and 50 cycles and a long tail extending to 270 cycles. This reflects a dataset dominated by late-stage degradation samples, with additional variability in machine lifespans across the fleet. Such skewness suggests that nonlinear models or transformations may be more appropriate than standard linear regression methods. The failure label distribution (Figure 2) further reinforces the need for careful modeling, as naive accuracy-based approaches may be misleading due to class imbalance. Evaluation metrics such as recall and F1-score are more appropriate, and techniques such as class weighting, SMOTE, or stratified cross-validation are recommended to ensure robust learning of minority failure cases.

Analysis of degradation behavior shows clear relationships between sensor variables and RUL (Figure 3).

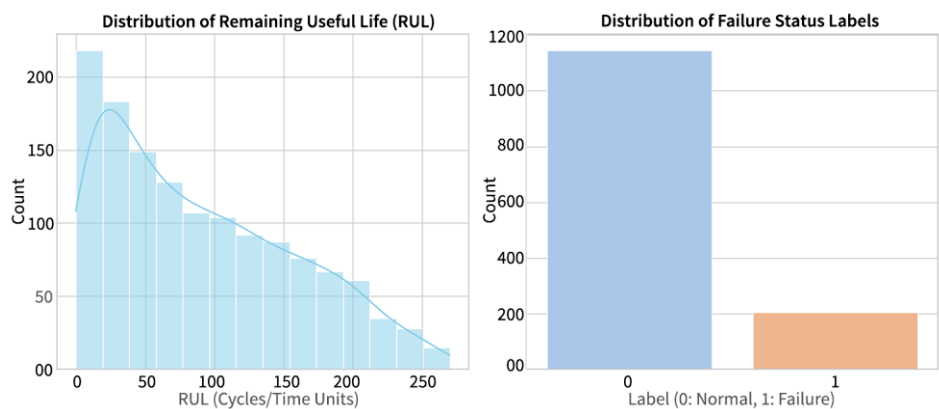


Figure 2. RUL and failure label distribution

Wear_last, temp_mean, vib_mean, and current_mean exhibit strong negative correlations with RUL (≈ -0.79), while pressure_mean shows a strong positive correlation ($\approx +0.79$), reflecting consistent and opposing trends during equipment deterioration. These variables collectively represent the primary degradation signals of the system.

Marked multicollinearity is observed among temperatures, vibration, and current measurements ($\approx +0.99$), alongside their strong inverse relationship with pressure_mean (≈ -0.99). This indicates substantial redundancy across sensor channels, which may justify dimensionality reduction in simpler models, although ensemble-based methods are generally more robust to such dependencies.

A comparison of feature types indicates that aggregated statistics are more informative than extreme values. Temp_mean shows a stronger relationship with RUL (-0.79) and a moderate association with failure ($+0.38$) compared to temp_max (-0.61 and $+0.28$), indicating that sustained operating conditions provide more reliable degradation information than short-term fluctuations. In contrast, temp_std, vib_std, and pressure_std exhibit negligible correlation with the targets, indicating limited predictive value. Overall, the feature set contains a small number of highly informative variables, substantial redundancy among correlated sensors, and several weak descriptors, supporting selective feature engineering.

Overall, the dataset reflects a coherent degradation process in which increasing thermal, mechanical, and electrical stress is associated with decreasing pressure and RUL.

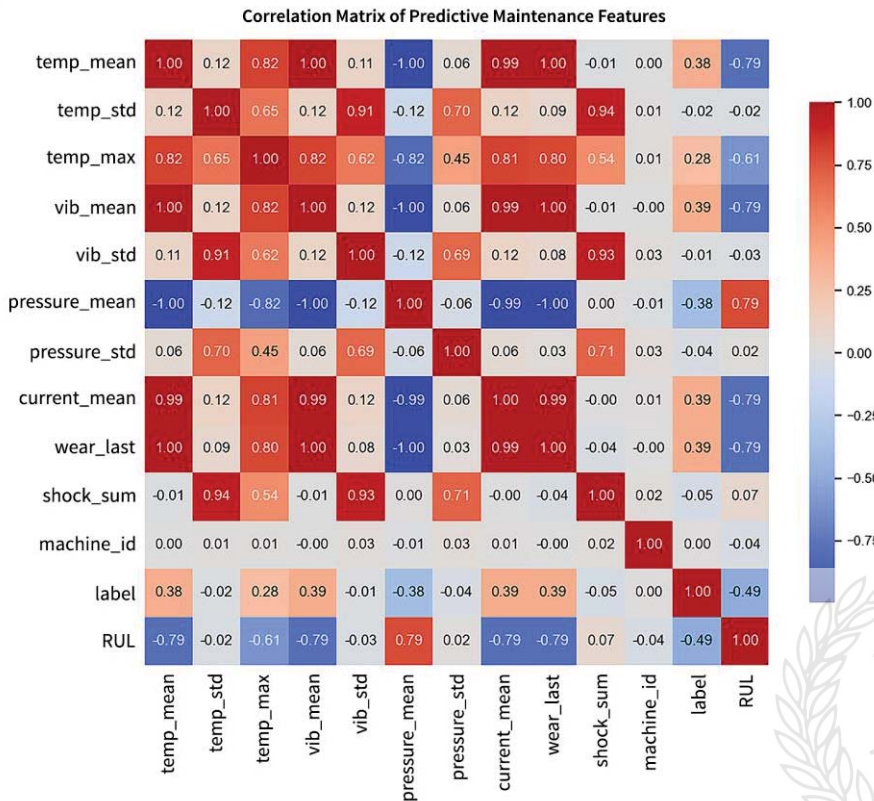


Figure 3. Correlation matrix

This structured behavior, together with clear class separability and strong inter-feature relationships, indicates that the dataset is well suited for predictive maintenance and failure prediction tasks.

3.3. Predictive Maintenance Framework

This study proposes a unified predictive maintenance framework for multivariate time-series sensor data. The framework integrates classical ML, deep learning models, anomaly detection methods, and XAI techniques to support both failure prediction and RUL estimation. The hybrid design enables accurate classification of machine states, reliable degradation modeling, real-time risk assessment, and interpretable decision-making. By combining complementary modeling approaches, the framework improves predictive robustness while ensuring transparency in industrial applications. The framework was developed in Python using several scientific and ML libraries, including NumPy, Pandas, Scikit-learn, XGBoost, TensorFlow/Keras, SHAP, Matplotlib, and Seaborn. These modules

were employed for data generation, preprocessing, feature engineering, model development, evaluation, visualization, and explainability analysis.

The proposed methodology is structured into five main components: dataset preparation, feature engineering, model development, evaluation, and real-time simulation. The experimental dataset comprises multivariate industrial sensor signals, including temperature, vibration, pressure, current, and machine wear indicators. Two predictive tasks are defined: (i) binary failure prediction and (ii) regression-based RUL estimation. The dataset is organized such that input features are derived from sensor measurements, while target variables correspond to failure labels and RUL values. The dataset was partitioned into training and testing subsets using an 80:20 observation-level split. For the classification task, stratified sampling was applied to preserve the proportion of the two failure classes in the training and testing subsets. The split was performed independently of machine_id; consequently, observations from the same machine may be present in both the training and testing subsets. This should be considered when interpreting the reported results, as machine-specific patterns may be partially shared between the two subsets. A machine-wise split based on machine_id would provide a stricter evaluation of model generalization to previously unseen machines and is therefore identified as an important direction for future validation. The experimental pipeline uses a unified engineered feature representation for conventional ML modeling, SHAP-based interpretability analysis, and real-time streaming inference. Raw sensor observations are transformed into ten engineered sensors and condition features derived from temperature, vibration, pressure, current, wear, and vibration-based shock measurements. Within each window, statistical descriptors are extracted to capture temporal dynamics and degradation behavior. These include the mean, standard deviation, and maximum values of temperature, vibration, and pressure signals, as well as aggregated vibration energy and the most recent machine wear state. This transformation converts sequential sensor data into structured feature vectors suitable for supervised learning algorithms. The machine_id variable is retained solely as an identifier and is excluded from model training. All benchmark models are trained using the same engineered predictor representation to ensure a fair and consistent comparison.

A range of supervised learning models is evaluated for both predictive tasks. For failure classification, the models include XGBoost, RF, GB, LogR, SVM, and KNN. For RUL estimation, XGBoost Regressor, RF Regressor, GB Regressor, LR, SVR, and KNN Regressor are employed.

Model performance is assessed using accuracy, F1-score, and ROC-AUC for classification, and mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2) for regression. Models are ranked based on

F1-score for classification and MAE for regression, and the best-performing models are selected for further analysis. Receiver operating characteristic curves are used to compare classification performance, while bar plots are utilized to summarize regression results across evaluation metrics.

For failure prediction, probabilistic outputs from the selected classifier are evaluated using precision–recall analysis. The Precision–Recall curve is used to assess the trade-off between precision and recall across different classification thresholds. The decision threshold is selected by maximizing the F1-score across candidate values.

A composite risk score is introduced to combine the predicted probability of failure with the estimated Remaining Useful Life (RUL) for continuous maintenance risk assessment. The risk score is defined as

$$R = 0.7P_f + 0.3\exp\left(-\frac{RUL}{50}\right), \quad (1)$$

where R denotes the composite risk score, P_f is the predicted probability of failure, and RUL is the predicted remaining useful life in cycles. The coefficient 0.7 assigns a greater contribution to the predicted failure probability, while the coefficient 0.3 incorporates the contribution of the estimated RUL . The exponential term provides a smooth decreasing function of RUL : as the predicted RUL decreases, $\exp(-RUL/50)$ increases, resulting in a higher contribution to the overall risk score. The value 50 represents the RUL scaling parameter used in the implementation. This formulation therefore combines immediate failure likelihood with remaining operational lifetime, giving greater emphasis to failure probability while increasing the risk contribution when the predicted remaining lifetime is short. In the real-time monitoring procedure, a maintenance alert is triggered when the composite risk score exceeds 0.7. A real-time streaming simulation environment is implemented to replicate industrial monitoring conditions. Synthetic sensor streams are generated using probabilistic distributions that reflect operational variability. At each time step, engineered features are computed and fed into trained models to estimate failure probability and RUL . The resulting risk score is continuously updated, and maintenance alerts are triggered when the risk exceeds a predefined threshold.

An unsupervised autoencoder is employed as a reconstruction-based anomaly scoring component. The model consists of fully connected encoder and decoder layers and is trained using the Adam optimizer with mean squared error (MSE) loss. The reconstruction error is calculated for each observation as the mean squared difference between the original input and its reconstructed representation and is used as an anomaly score. Higher reconstruction errors indicate

observations that are less consistent with the patterns learned by the autoencoder. In the present implementation, the reconstruction error is used as a continuous anomaly score; an explicit anomaly threshold and a separate quantitative anomaly-detection evaluation are not defined. Therefore, the autoencoder results are interpreted as reconstruction-based anomaly scoring rather than as a calibrated binary anomaly detector. Model interpretability is achieved using SHAP. A Kernel SHAP explainer is applied to the XGBoost classifier to quantify feature contributions to failure predictions. A representative background dataset consisting of 100 randomly selected observations from the training set is used to initialize the explainer. SHAP values are then computed for 200 observations from the test set. The feature space and feature ordering are kept identical between the training and test representations to ensure consistency during explanation. Both global and local explanations are generated, allowing identification of the most influential predictor variables in failure prediction.

Finally, a Transformer-based multi-task learning architecture is developed to jointly model multiple predictive maintenance objectives. The architecture incorporates positional encoding and stacked self-attention layers to capture temporal dependencies in sensor data. To quantitatively evaluate the Transformer component, the two model variants, Functional_1 and Functional_2, were additionally evaluated using a machine-wise train-test partition, with 40 machines used for training and 10 previously unseen machines used for testing. The machine_id variable was used only to define machine-wise data partitions and was not included as an input feature to the Transformer models. Because each machine contains 27 observations, whereas the Transformer architecture requires a sequence length of 30, the first observation of each machine was repeated for initial padding, ensuring that all sequence elements originated from the same machine. Functional_1 was evaluated for failure classification, whereas Functional_2 was evaluated for failure classification, RUL estimation, sequence reconstruction, and survival-oriented prediction. A shared representation layer feeds multiple task-specific output heads for failure classification, RUL regression with uncertainty estimation, anomaly reconstruction, and survival probability prediction. The model is trained using a weighted multi-objective loss function to balance the contributions of the individual tasks.

4. Results and Discussion

4.1. Failure Classification

The classification benchmark results for all tested ML models are presented in Table 1.

Table 1. Classification benchmark

Model	Accuracy	F1-score	ROC-AUC
XGBoost	0.859	0.441	0.833
RF	0.878	0.441	0.815
KNN	0.870	0.407	0.822
GB	0.852	0.333	0.834
LogR	0.837	0.043	0.815
SVM	0.848	0.000	0.592

The classification results summarized in Table 1 show that all evaluated models achieve relatively high accuracy, ranging from approximately 0.837 to 0.878. However, the substantially lower F1-scores indicate that this performance is influenced by class imbalance, where majority-class dominance inflates accuracy and masks poor minority-class detection. As a result, accuracy alone is not a reliable indicator of model effectiveness in this setting.

In terms of class separability, measured by ROC-AUC, GB achieves the highest score (0.834), closely followed by XGBoost (0.833). KNN also shows competitive performance (0.822), while RF and LogR are slightly lower but comparable. In contrast, SVM performs poorly (0.592), indicating limited ability to separate the classes. A clearer difference emerges in F1-score, which better reflects minority-class performance. XGBoost and RF achieve the best results (0.441), followed by KNN (0.407). GB drops to 0.333, while LogR and SVM perform poorly, with SVM failing to detect the minority class (F1 = 0). This confirms the limited suitability of linear decision boundaries for this dataset.

Overall, XGBoost provides the most balanced performance across both discrimination (ROC-AUC) and minority-class sensitivity (F1-score). Its advantage can be attributed to its gradient-boosted tree structure, which sequentially corrects errors made by previous learners while optimizing a regularized objective function. This enables better handling of complex, nonlinear relationships and subtle decision boundaries present in the sensor-derived feature space. In addition, built-in regularization and shrinkage help reduce overfitting, which is particularly beneficial under class imbalance. RF follows closely due to its robustness through bagging, while KNN remains moderately effective but less stable in high-dimensional feature spaces.

These results suggest that ensemble tree-based methods are best suited for this dataset. However, the uniformly low F1-scores across models highlights the impact of class imbalance. Addressing this through resampling strategies or class-weighted learning would likely improve minority-class detection and overall robustness.

For failure prediction, the precision-recall analysis provides additional insight into the classifier's performance under class imbalance. For the XGBoost classifier, the resulting AUPRC is 0.449, while the optimal classification threshold is 0.239. At this threshold, the classifier achieves a precision of 0.481, a recall of 0.610, and an F1-score of 0.538. The corresponding Precision–Recall curve is presented in Figure 4. The marked point on the curve represents the threshold that maximizes the F1-score.

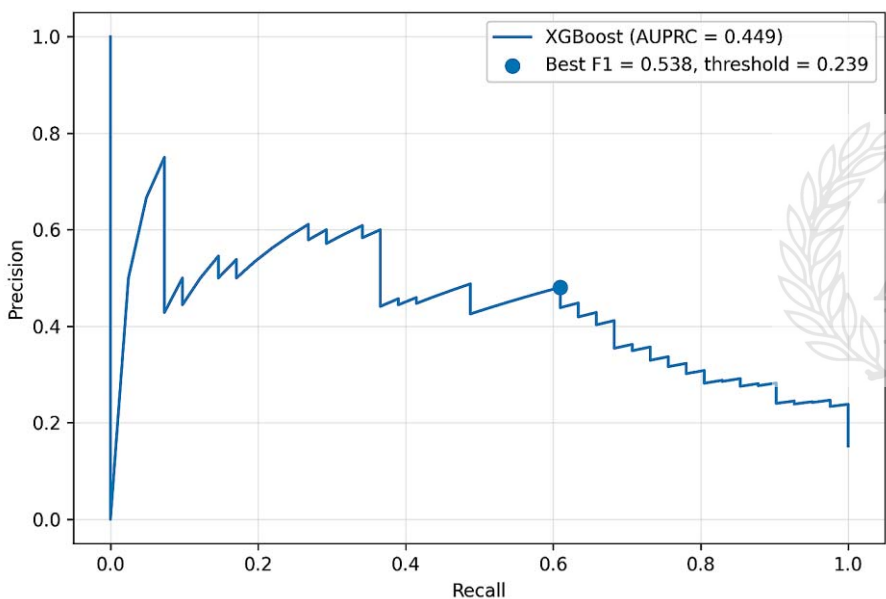


Figure 4. Precision-Recall curve for the XGBoost classifier

4.2. Remaining Useful Life Prediction

The RUL regression results show a clear hierarchy in model performance across all evaluation metrics (MAE, RMSE, and R^2), as illustrated in Figure 5. Overall, XGBoost achieves the best performance by a significant margin, followed by GB, while SVR performs the worst.

XGBoost records the lowest MAE (17.79) and RMSE (24.15), along with the highest R^2 score (0.8788), indicating both low prediction error and strong explanatory power, capturing nearly 88% of the variance in RUL. This suggests

that the model effectively learns complex nonlinear relationships in the data. GB follows with higher errors (MAE = 22.46, RMSE = 29.55) and a lower R^2 score (0.8184), remaining strong but clearly less accurate, reflecting the advantage of XGBoost's more optimized boosting and regularization strategy. KNN and RF show comparable performance. KNN achieves a slightly lower MAE (26.22) than RF (26.82), while RF performs marginally better in RMSE. Their R^2 values ($\approx 0.73 - 0.74$) indicate moderate predictive capability but limited ability to capture more complex nonlinear patterns compared to boosting-based models. LR performs worse, with a MAE of 31.91 and an R^2 of 0.6489, indicating that linear assumptions are insufficient for modeling the underlying structure of the data. SVR exhibits the weakest performance, with a high MAE (48.96), RMSE (61.92), and a low R^2 score (0.2029), explaining only about 20% of the variance. This suggests that the model is unable to effectively capture the relationships present in the feature space under the current configuration.

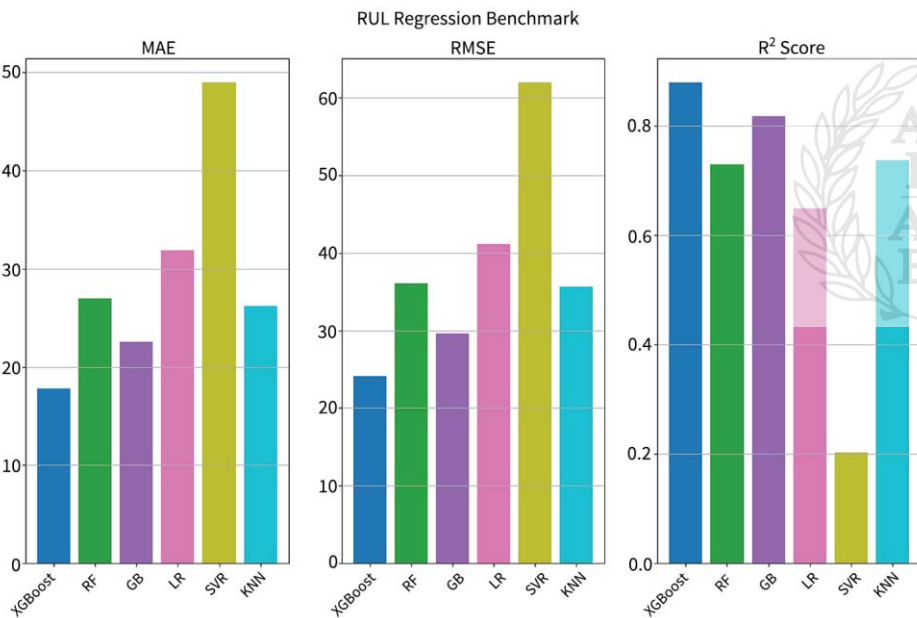


Figure 5. RUL regression results

Overall, the results favor tree-based ensemble methods. XGBoost achieves the best performance across all evaluated RUL regression metrics, while classification performance is more balanced, with XGBoost and Random Forest achieving the highest F1-score and Gradient Boosting achieving the highest ROC-AUC.

4.3. Comparative Performance Analysis Of Machine Learning Models

Across both classification and regression experiments, a consistent performance hierarchy is observed, with ensemble tree-based methods substantially outperforming linear models and SVM-based approaches. This pattern indicates that the underlying feature space is governed by nonlinear interactions and feature dependencies that are not adequately captured by simpler parametric formulations.

In the classification task, the relatively high accuracy values across all models contrast sharply with the substantially lower F1-scores, confirming the presence of class imbalance and highlighting the limited informativeness of accuracy as a standalone metric. XGBoost demonstrates the most balanced performance, achieving the highest F1-score alongside strong ROC-AUC, suggesting effective discrimination of minority failure events while maintaining overall class separability. RF exhibits comparable F1 performance, although with slightly reduced discriminative capability as reflected in ROC-AUC. GB remains competitive but shows reduced robustness in minority-class detection. KNN provides moderate performance, indicating partial sensitivity to local structure but limited generalization capacity in higher-dimensional feature space. In contrast, LogR and SVM exhibit clear limitations, with LogR failing to adequately capture minority-class structure and SVM demonstrating poor separability, likely due to insufficient alignment between kernel assumptions and the underlying data distribution.

A similar ranking emerges in the regression setting, where boosting-based methods again dominate performance. XGBoost achieves the lowest prediction error and highest R^2 , indicating a strong capability to capture complex nonlinear relationships between the predictor variables and RUL. This suggests that its iterative residual learning strategy, combined with regularization, provides a more effective bias–variance trade-off than competing methods. GB follows as a strong secondary model, although with reduced accuracy, implying that additional optimization in XGBoost contributes meaningfully to performance gains. RF and KNN occupy an intermediate position, capturing a substantial portion of variance but failing to model finer-grained nonlinear degradation dynamics. LR performs significantly worse, reinforcing the inadequacy of linear assumptions for this problem. SVR exhibits the weakest performance, indicating poor kernel alignment and limited capacity to generalize under the present feature representation.

From a unified perspective, the results indicate that ensemble tree-based methods provide strong performance for the considered predictive-maintenance tasks. XGBoost provides the strongest performance for RUL regression and achieves a competitive classification performance, while Random Forest and Gradient Boosting achieve the best results for individual classification metrics. These findings suggest that tree-based ensemble methods are well suited to the nonlinear

relationships present in the considered feature space, although the results do not establish universal superiority of a single model across all tasks and metrics. RF offers a stable but less expressive alternative, while KNN serves primarily as a locality-based baseline without competitive predictive strength. In contrast, linear models and SVM fail to adequately represent the structure of the data without substantial feature transformation or re-engineering.

Overall, the empirical evidence supports tree-based ensemble methods as strong baseline models for this predictive-maintenance problem, with XGBoost providing the strongest performance for the evaluated RUL regression task.

4.4. Cyclic Temporal Behavior of Sensor Signals

Figure 6 illustrates sensor evolution over the final ~200 cycles and reveals a strongly periodic operating regime characteristic of a cyclic industrial process rather than steady-state behavior. The repeated sawtooth profiles and abrupt resets across all signals indicate well-defined cycle boundaries, likely driven by batch operations, scheduled maintenance, or automated system restarts. This suggests that the system does not evolve continuously over time, but instead undergoes repeated operational and reset phases.

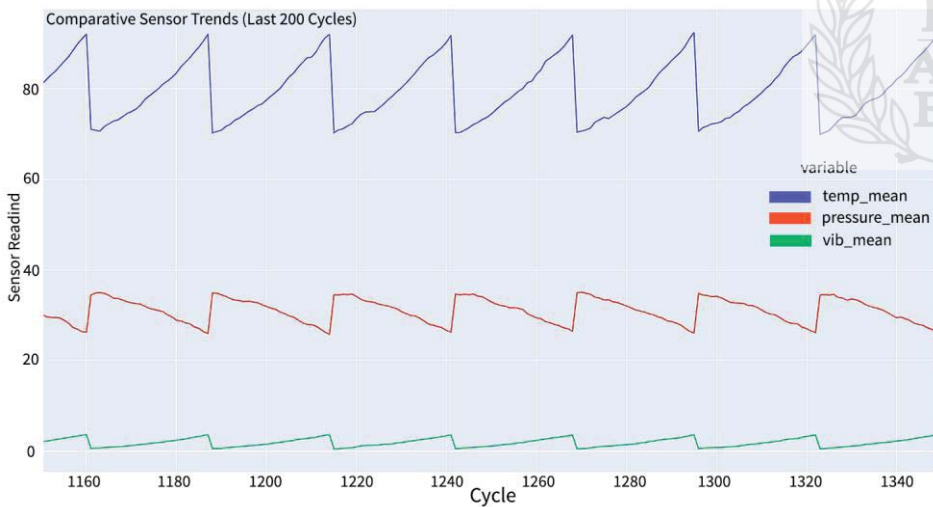


Figure 6. Sensor readings

Within each cycle, `temp_mean` increases smoothly from about 70 to 92 before abruptly returning to its initial level, reflecting continuous thermal energy buildup during operation followed by rapid cooling or reset. In contrast, `pressure_mean` follows an inverse pattern, gradually decreasing from approximately 35 to 26.

This opposing behavior is consistent with coupled thermodynamic effects, where increasing load or flow resistance leads to progressive pressure drop as thermal conditions intensify. The `vib_mean` signal shows a comparable cyclic trend to temperature, starting near zero after each reset and increasing steadily until cycle completion. This behavior reflects the gradual accumulation of mechanical stress, friction, or wear during operation. The alignment between rising temperature and vibration further suggests a shared dependence on system load, while periodic resets indicate restoration of baseline operating conditions.

Overall, the observed patterns indicate a tightly coupled thermomechanical system governed by cyclic operation. The consistent inverse relationship between temperature and pressure, together with synchronized increases in vibration, reflects structured operational loading and repeatable degradation behavior within each cycle.

The synchronized resets observed across all three signals are particularly significant. The simultaneous drops in temperature and vibration, together with the concurrent increase in pressure, suggest a deliberate system reset rather than a naturally occurring decay process. This pattern is consistent with an automated operational cycle in which system states are periodically restored to baseline conditions, indicating a controlled batch process or maintenance-driven operation rather than stochastic fluctuations or failure-induced transitions.

4.5. Streaming Framework

The streaming module simulates a real-time predictive maintenance environment through sequential generation of synthetic sensor observations over a fixed time horizon. Each observation includes key operational variables (temperature, vibration, pressure, current, and wear), sampled using probabilistic distributions to approximate realistic industrial operating conditions. A short inter-sample delay is introduced to emulate real-time data acquisition.

Incoming observations are converted into structured feature representations to ensure compatibility with trained models and enable consistent inference in a streaming setting. Two pre-trained models are applied: a classifier for estimating failure probability and a regression for predicting RUL. Their outputs are combined to compute a unified risk score that jointly reflects failure likelihood and degradation severity.

The system continuously monitors failure probability, RUL, and the derived risk score, triggering a maintenance alert when the risk exceeds a predefined threshold. The complete process is illustrated in Figure 7.

```
=== Streaming started ===
```

[1/20]	Risk: 0.052	Prob: 0.017	RUL: 101.4
[2/20]	Risk: 0.031	Prob: 0.018	RUL: 138.5
[3/20]	Risk: 0.068	Prob: 0.018	RUL: 84.7
[4/20]	Risk: 0.093	Prob: 0.037	RUL: 75.1
[5/20]	Risk: 0.035	Prob: 0.012	RUL: 120.3
[6/20]	Risk: 0.052	Prob: 0.031	RUL: 114.2
[7/20]	Risk: 0.035	Prob: 0.003	RUL: 110.4
[8/20]	Risk: 0.031	Prob: 0.005	RUL: 119.9
[9/20]	Risk: 0.036	Prob: 0.004	RUL: 109.6
[10/20]	Risk: 0.043	Prob: 0.017	RUL: 113.6
[11/20]	Risk: 0.035	Prob: 0.003	RUL: 110.2
[12/20]	Risk: 0.036	Prob: 0.005	RUL: 110.9
[13/20]	Risk: 0.078	Prob: 0.056	RUL: 102.4
[14/20]	Risk: 0.028	Prob: 0.007	RUL: 128.0
[15/20]	Risk: 0.033	Prob: 0.006	RUL: 117.6
[16/20]	Risk: 0.030	Prob: 0.008	RUL: 124.3
[17/20]	Risk: 0.042	Prob: 0.006	RUL: 103.3
[18/20]	Risk: 0.038	Prob: 0.005	RUL: 108.7
[19/20]	Risk: 0.045	Prob: 0.006	RUL: 99.7
[20/20]	Risk: 0.031	Prob: 0.007	RUL: 121.6

Figure 7. Real time streaming simulation results



The proposed streaming framework demonstrates the integration of real-time sensor monitoring, machine learning inference, and dynamic risk assessment within a predictive maintenance context. This setup supports continuous evaluation of machine health and enables timely decision-making for preventive maintenance in Industry 4.0 environments.

4.6. Transformer-Based Multi-Task Predictive Maintenance Architecture

A Transformer-based framework is proposed for predictive maintenance using multivariate sensor time-series data. The architecture uses a sequence length of 30 time steps and a 10-dimensional input representation consisting of ten engineered sensor and condition features. The machine_id variable is not used as a predictive input feature; instead, it is retained solely for identifying individual machines, grouping observations into machine-specific sequences, and supporting machine-wise data partitioning. The input is projected into a 64-dimensional embedding space, followed by sinusoidal positional encoding and stacked Transformer encoder blocks. These blocks incorporate multi-head self-attention, feed-forward

layers, residual connections, and normalization to model temporal dependencies in degradation patterns. Global average pooling is used to obtain a compact sequence representation that is shared across task-specific output heads. The architecture supports multiple predictive maintenance objectives, including binary failure classification, RUL regression with uncertainty estimation, sequence reconstruction for anomaly detection, and survival analysis for future risk estimation. The model is configured with the Adam optimizer and a weighted multi-task loss, with task-specific loss weights used to balance the individual learning objectives.

Within this framework, two model variants are introduced: Functional_1 and Functional_2. Functional_1 is a compact, single-task attention-based model focused on failure classification. It employs a lightweight attention-based architecture followed by global pooling and a dense output head, with approximately 3.8K parameters. Its reduced complexity enables fast inference and low computational demand, making it suitable for real-time and resource-constrained environments, although its modeling capacity is limited to relatively shallow temporal dependencies. In contrast, Functional_2 is a deeper Transformer-based multi-task architecture built on stacked encoder blocks, with approximately 292K parameters. It incorporates task-specific outputs for failure prediction, probabilistic RUL estimation, sequence reconstruction, and survival analysis, providing a unified architecture for modeling degradation dynamics, uncertainty, and anomaly-related behavior.

In the present implementation, both architectures are defined and compiled as components of the proposed framework. Their quantitative evaluation is performed using a separate machine-wise train-test procedure, with 40 machines used for training and 10 previously unseen machines used for testing. The results of the conventional ML benchmark are reported in Table 1, whereas Table 2 summarizes the performance of the two Transformer configurations. Functional_1 is evaluated only as a classification model, whereas Functional_2 represents the multi-task Transformer configuration and is evaluated using both classification and RUL regression metrics.

Table 2. Quantitative performance of the Transformer variants

Model	Accuracy	F1-score	ROC-AUC	MAE	RMSE	R ²
Functional 1	0.748	0.541	0.716	-	-	-
Functional 2	0.756	0.588	0.821	37.87	49.74	0.346

The results in Table 2 indicate that Functional_2 provides consistently better classification performance than Functional_1, particularly in terms of F1-score and ROC-AUC. Moreover, only Functional_2 was evaluated for RUL estimation,

demonstrating moderate agreement between the predicted and true RUL values. Figure 8 further illustrates this relationship.

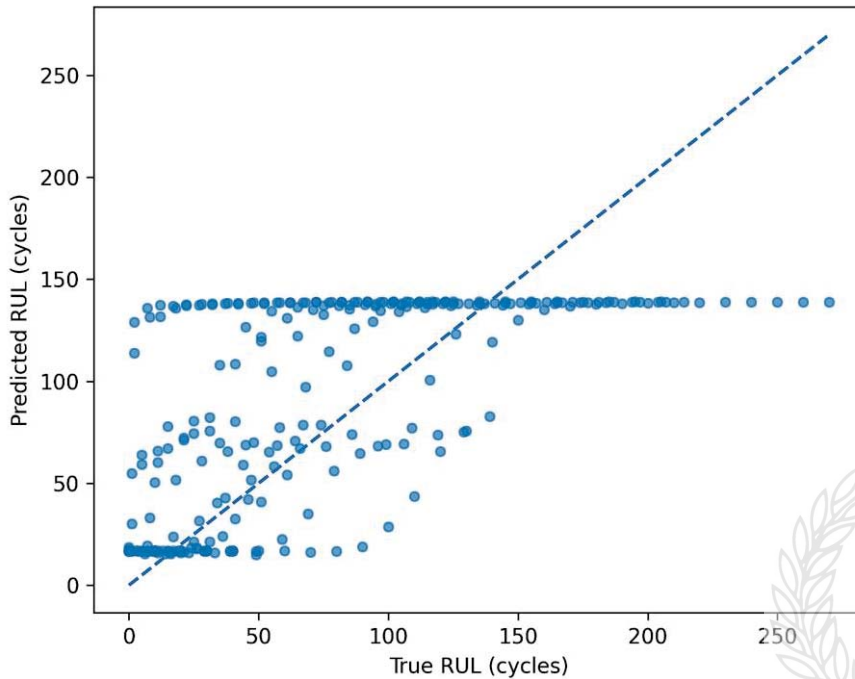


Figure 8. True versus predicted RUL values for Functional_2 model

The Precision-Recall curves for the two Transformer variants are presented in Figure 9, with Functional_2 demonstrating better overall performance than Functional_1.

The auxiliary outputs of Functional_2 were also quantitatively evaluated. The reconstruction component achieved a mean reconstruction MSE of 0.485, with an AUROC of 0.662 and an AUPRC (Area Under the Precision-Recall Curve) of 0.319 when the failure label was used as an auxiliary reference for anomaly-oriented evaluation. The survival-oriented output achieved a mean Brier score of 0.092 across the evaluated prediction horizons. These results provide additional quantitative evidence for the multi-task formulation, while the anomaly and survival outputs are considered supplementary to the primary classification and RUL objectives. The distribution of the reconstruction errors obtained by Functional_2 is shown in Figure 10.

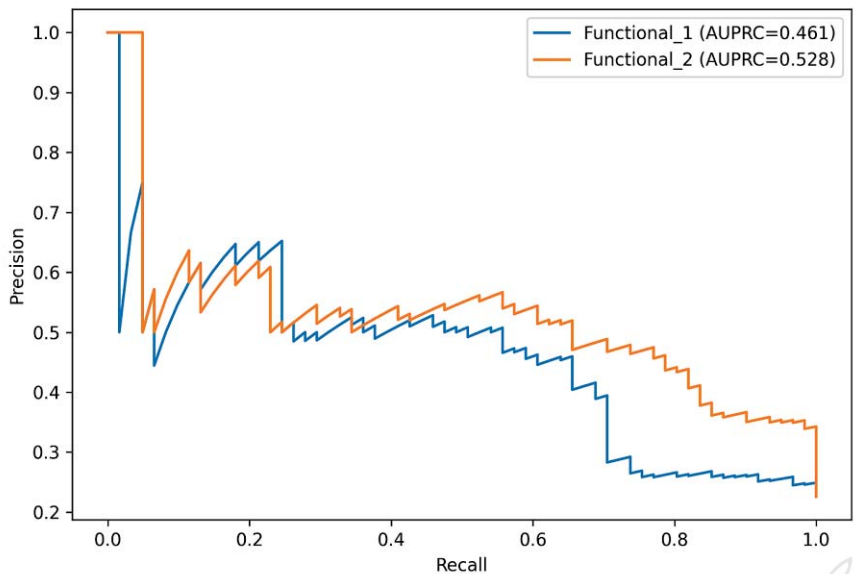


Figure 9. Precision-Recall curves for Functional_1 and Functional_2 models

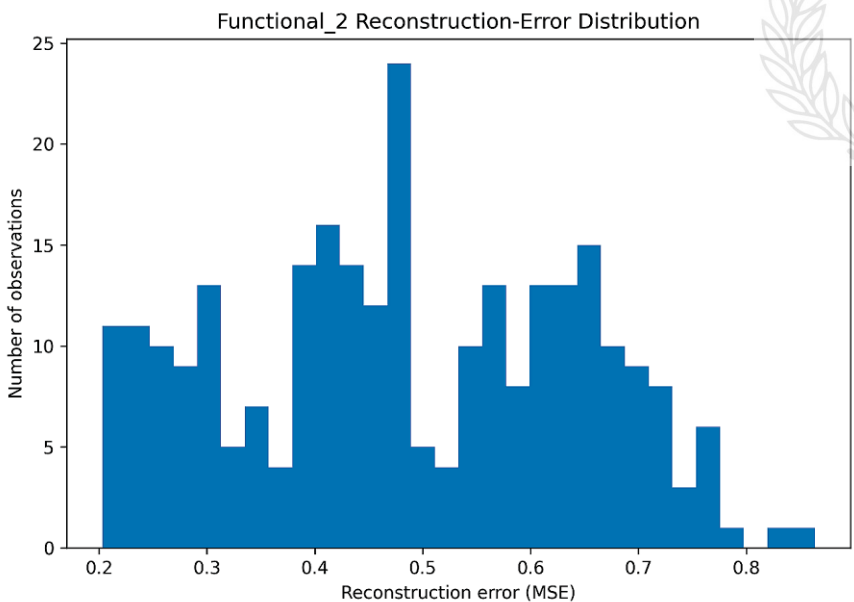


Figure 10. Distribution of reconstruction errors for Functional_2 model

Overall, Functional_1 provides a more compact architecture oriented toward computational efficiency, whereas Functional_2 provides greater representational

capacity through its multi-task formulation. The results indicate that Functional_2 achieves stronger classification performance than Functional_1, while also providing RUL, reconstruction, and survival-oriented outputs within a unified architecture. These characteristics make Functional_1 a lightweight alternative, whereas Functional_2 provides a more comprehensive framework for multi-objective machine health assessment.

4.7. SHAP Analysis

The SHAP analysis is performed on the predictor variables used by the classification model. The machine_id variable is treated as an identifier rather than an operational measurement and is therefore omitted from the reported SHAP feature-importance visualization and ranking. The reported SHAP importance values consequently focus on the operational predictor variables. Figure 11 presents the global SHAP feature importance based on the mean absolute SHAP values across the dataset. Larger absolute SHAP values indicate a greater overall contribution to the model output. Unlike the SHAP beeswarm plot, which additionally illustrates the direction and distribution of feature effects at the individual-sample level, the bar plot focuses on the overall magnitude of feature contributions. The results indicate that the model is predominantly driven by two key variables: vib_mean and current_mean, which exhibit nearly identical and substantially higher mean absolute SHAP values than all other features. This suggests that vibration and electrical current measurements are the principal predictive signals within the model and account for the majority of its explanatory power. From an engineering standpoint, these variables likely represent the most critical operational indicators of system behavior and are highly sensitive to changes in operating conditions. A secondary group of features includes temp_mean and pressure_mean, followed by variability- and peak-related metrics such as pressure_std, temp_std, vib_std, and temp_max. Although these features contribute less strongly, they still provide meaningful information to the model. Their role appears to be primarily contextual, refining predictions rather than driving them. The presence of both mean-based and standard deviation-based features within this group indicates that the model leverages both steady-state operating conditions and dynamic fluctuations in system behavior. At the lower end of the importance scale, shock_sum and especially wear_last contribute very little to the model's overall behavior. The near-zero impact of wear_last confirms that it is effectively unused in the current training setup, while shock_sum provides only a minor contribution.

Overall, the feature importance distribution reveals a clear hierarchy: core predictive power is concentrated in real-time operational signals (vibration and current), while environmental and variability-related factors play a supportive role. Features with consistently low importance likely add limited incremental

value and may be candidates for simplification or removal with minimal impact on model performance.

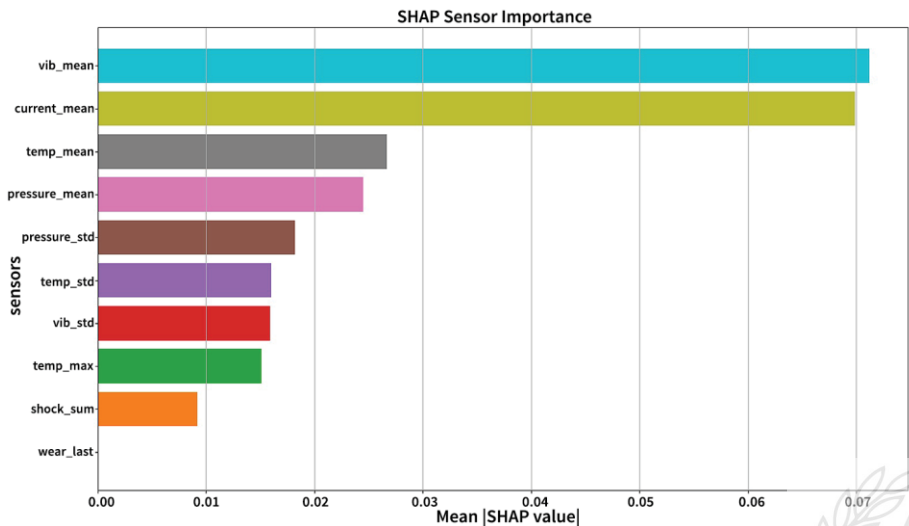


Figure 11. SHAP feature importance

The SHAP beeswarm plot (Figure 12) provides a detailed view of the importance and direction of the operational features. Features are ranked by their mean absolute SHAP values, with vib_mean and current_mean emerging as the dominant predictors. Higher values of these variables are generally associated with positive SHAP values, indicating an increased contribution to the model output. For both variables, higher values (red) are consistently associated with positive SHAP values, increasing the model output, while lower values (blue) reduce predictions, indicating a strong and largely monotonic relationship between these signals and the target variable. From an applied perspective, elevated vibration and current are therefore the most influential indicators of increased system risk or abnormal operating conditions.

Temp_mean also increases risk when high, though its effect is weaker and more concentrated. In contrast, pressure_mean shows a more mixed, non-linear relationship, with its impact varying across different value ranges. Shock_sum has limited influence under normal operating conditions, but higher values can still shift predictions noticeably. The least important features are temp_std, vib_std, and especially wear_last, which the model largely ignores. Overall, the model is primarily driven by average vibration and current as the strongest indicators of machine degradation, with temperature and pressure acting as secondary contributors.

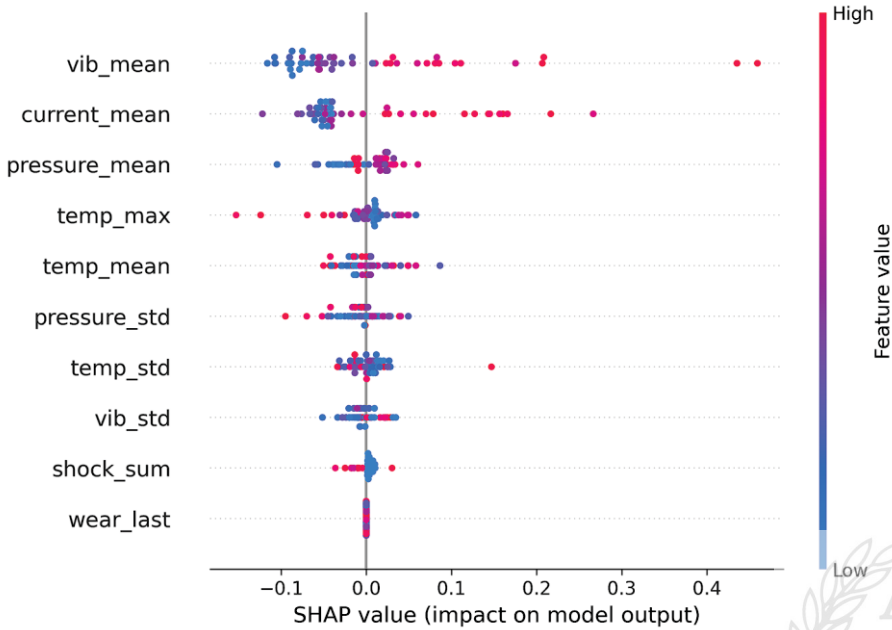


Figure 12. SHAP beeswarm plot

Linking the sensor time-series patterns (Figure 6) with the SHAP analysis provides a coherent explanation for the observed hierarchy of feature importance. The pronounced contribution of `vib_mean` and `current_mean` is consistent with their behavior as progressive cycle indicators, as both variables exhibit a steady increase within operational windows. This makes them highly informative for inferring cycle progression and potentially anticipating reset or transition events. In this context, their dominant SHAP values reflect their ability to encode temporal progression in a compact and highly predictive form. Likewise, the negative SHAP association observed for `pressure_mean` aligns with its characteristic downward trend over the course of each cycle. Lower pressure levels are indicative of later operational stages, and the model appears to exploit this relationship when forming its predictions. Depending on the target definition, these later-stage conditions may correspond to either increased or decreased predicted outcomes, but in both cases, pressure effectively functions as a temporal marker.

More broadly, these findings indicate that the model is not relying solely on static threshold-based relationships. Instead, it appears to capture underlying temporal structure within the data, leveraging gradually within-cycle changes in vibration,

temperature, and pressure as implicit indicators of system evolution. This suggests that the model encodes cycle progression indirectly through sensor drift patterns, thereby using time-dependent behavior as a key predictive signal rather than treating each observation as independent.

5. Conclusion

This study presented a hybrid XAI framework for PdM that integrates conventional ML models, Transformer-based temporal modeling, anomaly reconstruction, and SHAP-based interpretability within a unified analytical pipeline. The proposed framework is designed to support failure prediction, RUL estimation, real-time condition monitoring, and interpretable decision-making in IoT- and IIoT-enabled industrial environments.

The results show that ensemble ML methods provide the strongest overall predictive performance. XGBoost achieved the best results for RUL estimation, while RF and XGBoost achieved the highest F1-score for failure classification and GB obtained the highest ROC-AUC. These findings confirm the effectiveness of tree-based ensemble models for capturing nonlinear degradation patterns in industrial sensor data.

The multi-task Transformer extended the framework by jointly supporting failure prediction, RUL estimation, anomaly reconstruction, and survival-oriented modeling. Functional_2 achieved stronger classification performance than Functional_1, although its RUL estimation did not surpass the XGBoost regression benchmark. The results support the feasibility of the multi-task Transformer as a unified architecture for multiple PdM objectives.

SHAP analysis identified vibration- and electrical-current-related engineered features as the most influential predictors, improving model transparency and interpretability. Overall, the proposed framework demonstrates the complementary value of ensemble ML, Transformer-based temporal modeling, reconstruction-based analysis, and XAI for scalable and intelligent PdM in Industry 4.0 and Industry 5.0 environments.

Future research should validate the framework using large-scale real-world industrial datasets, address class imbalance, and further optimize the architecture for deployment in resource-constrained IoT and edge environments.

References

- [1] Fernandes, M., Corchado, J. M., and 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: 14246-14280. <https://doi.org/10.1007/s10489-022-03344-3>
- [2] Paredes, J., Chavez, D., Isa-Jara, R., and Vargas, D. (2026), Hybrid approaches of machine learning algorithms in predictive maintenance: a systematic literature review, *Applied System Innovation*, 9(5): 90. <https://doi.org/10.3390/asi9050090>.
- [3] Miraje, A. A., Hunnargi, N. S., Kolap, A. S., Bhangari, D. S., and Chougule, S. A. (2022), Machine learning applications for predictive maintenance in mechanical systems: case studies, algorithms, and performance evaluation, *International Journal on Recent and Innovation Trends in Computing and Communication*, 10(12): 254-263
- [4] Tsallis, C., Papageorgas, P., Piromalis, D., and Munteanu, R. A. (2025), Application-wise review of machine learning-based predictive maintenance: trends, challenges, and future directions, *Applied Sciences*, 15(9): 4898. <https://doi.org/10.3390/app15094898>
- [5] Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., and Safaei, B. (2020), Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0, *Sustainability*, 12(19): 8211. DOI: <https://doi.org/10.3390/su12198211>.
- [6] Bitam, T., Yahiaoui, A., Boubiche, D. E., Martínez-Peláez, R., Toral-Cruz, H., and Velarde-Alvarado, P. (2025), Artificial intelligence of things for next-generation predictive maintenance, *Sensors*, 25(24): 7636. <https://doi.org/10.3390/s25247636>
- [7] Amin, A. A., Mubarak, A., Waseem, S., Alqarni, Z. A., and Manzoor, H. U. (2026), Remaining Useful Life (RUL) prediction methods for machine health estimation and fault diagnosis: a comprehensive review of latest techniques and future prospects, *Engineering Reports*, 8(4): e70699. <https://doi.org/10.1002/eng2.70699>
- [8] Amaliah, N. R., Tjahjono, B., and Palade, V. (2025), Human-in-the-loop XAI for predictive maintenance: A systematic review of interactive systems and their effectiveness in maintenance decision-making, *Electronics*, 14(17) : 3384. <https://doi.org/10.3390/electronics14173384>
- [9] Cummins, L., Sommers, A., Ramezani, S. B., Mittal, S., Jabour, J., Seale, M., and Rahimi, S. (2024), Explainable predictive maintenance: a survey of current methods, challenges and opportunities, *IEEE Access*, 12: 57574-57602. <https://doi.org/10.1109/ACCESS.2024.3391130>

- [10] Gopalan, R., Onniyil, D., Viswanathan, G., and Samdani, G. (2025), Hybrid models combining explainable AI and traditional machine learning: A review of methods and applications, *World Journal of Advanced Engineering Technology and Sciences*, 15(02): 1388-1402. <https://doi.org/10.30574/wjaets.2025.15.2.0635>
- [11] Mavrelis, I., Chatzopoulos, A., and Zacharia, P. (2026), Hybrid deep learning for predictive maintenance in industrial machinery using LSTM and MLP models, *Machines*, 14(2) : 191. <https://doi.org/10.3390/machines14020191>
- [12] Endla, P., Bhardwaj, S., Mathiyalagan, P., Akila, K., Sanjeevkumar, P., and Srinivasulu, M. (2025), AI-driven predictive maintenance framework for smart manufacturing: real-time deployment, multi-sensor fusion and scalable efficiency optimization, In Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 5, pp. 782-789, SCITEPRESS - Science and Technology Publications, Lda. <https://doi.org/10.5220/0013943600004919>
- [13] Uzel, H. (2025), Machine Learning–Based Fault Prediction Using Industrial Sensor Data: A Data-Driven Approach. In: Arslan, S. (ed.) Elektrik-Elektronik ve Haberleşme Mühendisliğinde Güncel Araştırmalar ve Değerlendirmeler I. <https://doi.org/10.58830/ozgur.pub1128.c4649>
- [14] Yang, C.-L., Meles, T. Y., Yilma, A. A., and Teshome, M. M. (2026), Uncertainty aware predictive maintenance using a hybrid Transformer with Monte Carlo Dropout and conformal prediction, *Ain Shams Engineering Journal*, 17(2) : 103992. <https://doi.org/10.1016/j.asej.2026.103992>
- [15] Hrnjica, B., Softic, S. (2020), Explainable AI in manufacturing: a predictive maintenance case study. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds), *Advances in Production Management Systems. Towards Smart and Digital Manufacturing. APMS 2020. IFIP Advances in Information and Communication Technology*, vol 592. Springer, Cham, pp. 66-73. https://doi.org/10.1007/978-3-030-57997-5_8
- [16] Shrestha, S. (2026), A conceptual model for integrating NLP and hybrid intelligence in industrial condition monitoring and predictive maintenance, *International Journal of Engineering Research and Reviews*, 14(1): 53-60. <https://doi.org/10.5281/zenodo.19254596>
- [17] Al-Refaie, A., Al-Atrash, M., and Lepkova, N. (2025), Prediction of the remaining useful life of a milling machine using machine learning, *MethodsX*, 14: 103195. <https://doi.org/10.1016/j.mex.2025.103195>
- [18] Choo, Y.-S., and Shin, S.-J. (2025), Integrating machine learning-based remaining useful life predictions with cost-optimal block replacement for industrial maintenance, *International Journal of Prognostics and Health Management*, 16(1): 1-22. <https://doi.org/10.36001/IJPHM.2025.v16i1.4242>

- [19] Mishra, T., Agrawal, D., Sonkar, S. K., and Tyagi, N. (2025), Data-driven approaches to battery health monitoring: machine learning for RUL estimation and fault diagnosis, *International Journal of Creative Research Thoughts (IJCRT)*, 13(10) : b239–b248
- [20] Kang, Z., Catal, C., and Tekinerdogan, B. (2021), Remaining Useful Life (RUL) prediction of equipment in production lines using artificial neural networks, *Sensors*, 21(3): 932. <https://doi.org/10.3390/s21030932>
- [21] Xu, Z., and Zhang, Q. (2026), Predictive maintenance optimization for industrial equipment via reliable prognosis and risk-aware reinforcement learning, *Complex & Intelligent Systems*, 12(1): 11. <https://doi.org/10.1007/s40747-025-02127-w>
- [22] Shaheen, B., Kocsis, Á., and Németh, I. (2023), Data-driven failure prediction and RUL estimation of mechanical components using accumulative artificial neural networks, *Engineering Applications of Artificial Intelligence*, 119: 105749. <https://doi.org/10.1016/j.engappai.2022.105749>
- [23] Paredes, J., Chávez, D., Isa-Jara, R., and Vargas, D. (2025), A hybrid machine learning algorithm approach to predictive maintenance tasks: A comparison with machine learning algorithms, *Results in Engineering*, 26: 105137. <https://doi.org/10.1016/j.rineng.2025.105137>
- [24] Kavitha, M. (2025), Hybrid AI-mathematical modeling approach for predictive maintenance in rotating machinery systems, *Journal of Applied Mathematical Models in Engineering*, 1(1) : 1-8. <https://doi.org/10.17051/JAMME/01.01.01>
- [25] Noussis, A., O'Neil, R., Saif, A., et al. (2025), A hybrid Bi-LSTM model for data-driven maintenance planning, *Autonomous Intelligent Systems*, 5: 13. <https://doi.org/10.1007/s43684-025-00099-9>
- [26] Fan, Z., Li, W., and Chang, K.-C. (2023), A bidirectional long short-term memory autoencoder transformer for remaining useful life estimation, *Mathematics*, 11(24): 4972, <https://doi.org/10.3390/math11244972>
- [27] Hou, C., and Zheng, L. (2024), A multi-task joint learning model based on transformer and customized gate control for predicting remaining useful life and health status of tools, *Sensors*, 24(13): 4117. <https://doi.org/10.3390/s24134117>
- [28] Yang, H., Yang, X., Sun, D., and Hu, Y. (2025), A MID-1DC+LRT multi-task model for SOH assessment and RUL prediction of mechanical systems, *Sensors*, 25(5) : 1368. <https://doi.org/10.3390/s25051368>
- [29] Shadi, M. R., Mirshekali, H., and Shaker, H. R. (2025), Explainable artificial intelligence for energy systems maintenance: A review on concepts, current techniques, challenges, and prospects, *Renewable and Sustainable Energy Reviews*, 216: 115668. <https://doi.org/10.1016/j.rser.2025.115668>
- [30] Anđelić, N., Baressi Šegota, S., and Mrzljak, V. (2025), Hybrid explainable AI for machine predictive maintenance: from symbolic expressions to meta-ensembles, *Processes*, 13(7): 2180. <https://doi.org/10.3390/pr13072180>

- [31] Villagómez-Galindo, M., Manjula, G., Jagadeesh, B. N., Patil, M., Potharaju, S., and Prasad, N. A. (2026), A hybrid ensemble and explainable AI Framework for predictive maintenance of industrial equipment, *Journal of Artificial Intelligence and Technology*, 6: 328-336. <https://doi.org/10.37965/jait.2026.0939>
- [32] Siva, R., and Ayush, R. (2026), Time aware explainable AI for fault diagnosis, *International Journal for Multidisciplinary Research (IJFMR)*, 8(2): 854-858
- [33] Özcan, H. (2025), Interpretable ensemble remaining useful life prediction enables dynamic maintenance scheduling for aircraft engines, *Scientific Reports*, 15(1) : 39795. <https://doi.org/10.1038/s41598-025-23473-2>

