STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, December 2026, pp 5967–5992.
Published online in International Academic Press (www.IAPress.org)

Development of machine learning algorithms for predictive maintenance of equipment in the automotive industry

Dinara Aldasheva [1], Olga Salykova [2], Kalybek Maulenov [3,*], Petr Radchenko [4], Tatyana Radchenko[5]

[1]*Department of Information Technology and, Automation, Kostanay Engineering and, Economics University named after M. Dulatov, Republic of Kazakhstan*
[2]*A. Baitursynuly Kostanay Regional University, Republic of Kazakhstan*
[3]*Department of Software, A. Baitursynuly Kostanay Regional University, Republic of Kazakhstan*
[4]*Department of Information Systems, A. Baitursynuly Kostanay Regional University, Republic of Kazakhstan*
[5]*Department of Physics, Mathematics and Digital Technology, A. Baitursynuly Kostanay Regional University, Republic of Kazakhstan*

Abstract The aim of this study was to develop and evaluate a machine-learning pipeline for predictive maintenance of industrial equipment in the automotive industry. The empirical basis comprised twelve months of multisensor observations collected from fifteen production lines, including vibration, temperature, pressure, and process-fluid-flow measurements, together with maintenance and failure logs. The proposed framework combined aligned tabular and sequential representations of sensor data and integrated Random Forest, XGBoost, and LSTM predictions through soft-voting ensemble fusion under a time-based validation protocol. The results showed that the ensemble outperformed the best standalone model, increasing Accuracy from 0.93 to 0.95 and F1-score from 0.88 to 0.91, while achieving ROC AUC = 0.96 on the held-out test interval. Pilot deployment in online monitoring mode confirmed the operational relevance of the predictive signal and showed that the system can support threshold-based early warning in real production conditions. The economic assessment further indicated that predictive maintenance based on the proposed pipeline reduced overall maintenance costs by approximately 28% and unplanned downtime by about 41%. These findings suggest that predictive maintenance in automotive production should be interpreted not only as a classification task, but as an integrated decision-support problem requiring synchronised heterogeneous data, time-aware evaluation, and deployment-oriented alert logic.

Keywords Time Series, Transfer Learning, Internet of Things, Gradient Boosting, Ensemble Models, Condition Monitoring Sensors, Maintenance Optimisation, Operational Parameters

AMS 2010 subject classifications 68T05, 68T10, 90B25

DOI: 10.19139/soic-2310-5070-3227

1. Introduction

The modern automotive industry is undergoing profound transformations driven by the need to improve equipment reliability, reduce operational costs, and optimise maintenance processes. The increasing complexity of machinery and equipment, along with growing demands for their performance and dependability, compels enterprises to seek innovative approaches for enhancing the efficiency of their production processes. Traditional maintenance strategies, such as scheduled servicing or reactive repairs, are not always sufficiently effective, as they often involve downtime, excessive resource consumption, and high costs. In this context, predictive maintenance emerges as a promising paradigm, enabling a shift from reactive problem-solving towards forecasting and prevention.

*Correspondence to: Kalybek Maulenov (kalybekmaulenov1@gmail.com) Department of Software, A. Baitursynuly Kostanay Regional University. 47 Baytursynov Str., Kostanay, Republic of Kazakhstan (110000).

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

Predictive maintenance involves leveraging data collected from sensors installed on equipment to assess current conditions and predict probable failures. Shah [1] emphasises that machine learning (ML) algorithms based on sensor data can enhance prediction accuracy and reduce costs through early detection of potential faults in autonomous vehicles. Recent work has also strengthened the case for automotive predictive maintenance under imbalanced failure conditions. Mahale et al. [2] showed that predictive-maintenance performance in automotive settings depends strongly on how class imbalance is handled during model design and evaluation, while de Giorgio et al. [3] demonstrated more broadly that imbalance is a systematic methodological challenge in manufacturing analytics.

Internet of Things (IoT) and ML technologies play a pivotal role in implementing such approaches, offering powerful tools for the analysis of large-scale datasets. Wan et al. [4] note that applying transfer learning techniques to heterogeneous data sources accommodates the complexity of operational environments and achieves higher fault diagnosis accuracy.

State-of-the-art data analysis algorithms are widely utilised for failure and fault prediction. Borré et al. [5] describe a hybrid Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) model with attention mechanisms, combining the precision of deep learning with the efficiency of time series analysis, rendering it highly suitable for equipment diagnostics tasks. Similarly, Jain et al. [6] in their systematic review confirm that ensemble methods such as gradient boosting play a crucial role in improving failure prediction accuracy.

One promising direction is the application of deep reinforcement learning. Chen and Wang [7] developed a maintenance planning approach for multi-component systems, accounting for complex interdependencies among components. This method proved effective under demanding operational conditions.

The role of big data in predictive maintenance also warrants attention. Estella [8] highlights those big data analytics not only improves prediction accuracy but also minimises operational costs by optimising maintenance schedules.

Finally, integrating predictive maintenance within the Industry 4.0 framework is a significant avenue. Taş [9] reports successful case studies demonstrating these approaches' importance in enhancing enterprise competitiveness, reducing risks, and optimising equipment performance.

A clear research gap remains in the current predictive-maintenance literature for automotive manufacturing. First, many existing studies focus either on tabular machine-learning models or on deep sequential architectures, but do not explicitly align heterogeneous multi-rate sensor streams into a unified decision pipeline. Second, current approaches often provide limited treatment of operational adaptation, including class imbalance, threshold selection, and robustness to changing production regimes. Third, relatively few studies are designed specifically for automotive production lines, where continuous multi-shift operation, high variability of vibration signatures, frequent load changes, and the safety-critical role of key assemblies create a more demanding environment than generic industrial monitoring scenarios. For this reason, the present study addressed not only model accuracy, but also synchronised tabular–sequential integration, time-aware validation, and deployment-oriented decision logic for automotive predictive maintenance.

To achieve this goal, the following tasks were set:

- analyse the specifics of the automotive industry in the context of equipment operating conditions and requirements for predictive maintenance systems;
- develop and test a suite of ML models, including Decision Trees, Random Forest, Gradient Boosting, and deep neural networks, employing transfer learning and ensemble approaches;
- evaluate the accuracy, robustness, and reliability of the developed algorithms on test datasets and in real-world production settings;
- conduct pilot testing of the proposed solutions to determine their practical effectiveness and economic feasibility;
- identify prospective directions for further algorithm development, considering the integration of additional data sources and increased model complexity.

2. Materials and methods

The study focused on industrial equipment at the SaryarkaAvtoProm Limited Liability Partnership (LLP), comprising fifteen production lines with various technological specialisations. Observations were carried out continuously over a twelve-month period, from January to December 2022, to capture all significant phases of the production cycle and account for seasonal fluctuations. Each line operated under variable loads and in a continuous multi-shift mode characteristic of the automotive sector. Sensors were installed on key equipment nodes (bearings, gearboxes, hydraulic cylinders) to measure vibration, pressure, temperature, and process fluid flow. Vibration sensors had a high sampling frequency, enabling detailed real-time capture of oscillations, whereas thermocouples and pressure gauges provided less frequent but stable readings. All measurements were time-stamped, facilitating alignment of heterogeneous signal types. In parallel, failure and maintenance logs were maintained, recording fault dates, downtime duration, fault types, and the list of performed repairs. This dataset enabled correlation of anomalous sensor signals with actual breakdowns, allowing ML algorithms to be trained on authentic production processes.

Raw multi-sensor data often contained missing values and noise arising from transmission errors or brief sensor outages. To mitigate such distortions, statistical filtering methods were applied: Outliers were handled using a fixed, reproducible interquartile range (IQR) rule applied per sensor channel. Values outside the bounds $[Q_1 - 1.5 \cdot \mathrm{IQR},\ Q_3 + 1.5 \cdot \mathrm{IQR}]$ were winsorised (clipped) to the nearest bound rather than removed, which reduced the risk of discarding genuine early degradation signatures that can manifest as extreme values shortly before failures. Thresholds were estimated on the training interval only and then applied unchanged to validation and test intervals. Missing values were imputed only for short gaps to preserve the true operational profile. Specifically, gaps up to 3 consecutive steps (i.e., up to 3 minutes for $\Delta t = 60$ s) were filled using linear interpolation, while longer gaps were not interpolated. Sliding windows with missingness above 10% across all channels were excluded from modelling to avoid creating artificial pre-failure patterns. Subsequently, timestamp synchronisation was performed, accounting for the differing sampling rates of devices: high-frequency vibration series were aggregated into metrics (mean, variance, maximum, and minimum), while slower parameters (temperature, pressure) were aligned to the same time steps. Features with substantially different scales were normalised using min–max scaling to the $[0, 1]$ range. For strictly positive and right-skewed signals, a $\log(1 + x)$ transformation was applied prior to scaling when needed. To prevent temporal leakage, all transformation parameters (log transform settings and scaling ranges) were estimated on the training interval only and then applied unchanged to the validation and test intervals. This pre-processing pipeline yielded a coherent dataset containing both tabular and temporal features, capturing the dynamics of wear and degradation. The dataset was then split into training and test subsets using a chronological (time-based) protocol: earlier time intervals were used for training, while the final time interval was reserved for independent testing, reflecting the real deployment scenario where models trained on historical data must predict future failures. Ensemble fusion used soft voting at the shared endpoint timestamp t, combining the aligned probabilities from Random Forest, XGBoost, and LSTM into a single risk score $p_{\mathrm{ens}}(t)$ used for alerting.

To combine tabular machine-learning models with deep time-series models, all inputs were aligned to a shared timeline and defined at a common endpoint timestamp. First, heterogeneous sensor streams were mapped to a unified time grid with step $\Delta t = 60$ s. High-frequency vibration measurements were aggregated to this grid by computing per-step descriptors over each interval $[t - \Delta t, t]$, including mean, variance, RMS, minimum, and maximum; slower parameters (temperature, pressure, and process/fluids flow) were aligned to the same timestamps using forward/nearest alignment.

Supervised samples were then constructed as sliding windows of fixed length L_{steps} with stride S_{steps}. For each window ending at timestamp t, two aligned representations were created: (i) a time-series tensor $\mathbf{X}_{\mathrm{seq}}(t) \in \mathbb{R}^{L \times F}$, where F is the number of step-level features on the unified grid, used as input to the LSTM; and (ii) an aggregated tabular vector $\mathbf{x}_{\mathrm{tab}}(t) \in \mathbb{R}^K$ computed over the same L-step window, including window-level statistics such as mean, standard deviation, minimum, maximum, trend, and variability measures for each sensor channel, used as input to Random Forest and XGBoost. The prediction target was defined as horizon-based failure anticipation. Each window endpoint t received a single binary label $y(t)$: $y(t) = 1$ if a log-confirmed failure occurred within the next prediction horizon $(t, t + H]$; otherwise $y(t) = 0$. This mapping ensured that both the tabular vector and

the time-series window corresponded to the same operational moment and the same prediction target. Missing or misaligned timestamps were handled by enforcing the unified grid: short gaps up to 3 consecutive steps were imputed using linear interpolation, while windows exceeding 10% missingness or containing long gaps were excluded to avoid distorting pre-failure patterns. If a sequential window was excluded due to excessive missingness, the corresponding timestamp t was not scored by the LSTM. Such timestamps were excluded from alert generation to avoid inconsistent ensemble fusion across model components. All train/validation/test splits were performed chronologically to prevent temporal leakage. The dataset composition, class distribution, sensor-channel breakdown, and chronological split intervals resulting from this construction are reported in Table 1. To avoid temporal leakage, the chronological split was applied before constructing supervised windows and aggregated features. Sliding windows were generated separately within each split interval, ensuring that no window crossed train/validation/test boundaries and that future timestamps were never used to form training inputs.

The model suite included a baseline Decision Tree, a Random Forest ensemble, and a gradient boosting classifier implemented as XGBoost for tabular features, as well as an LSTM network for sequential inputs. Hyperparameters were tuned using a time-based validation interval (chronological protocol) to prevent temporal leakage. For tree-based models, the final settings (e.g., depth limits, number of estimators, split/leaf constraints, and class-imbalance handling) are reported in Table 2. For the neural model, the layer-by-layer architecture, optimiser, learning-rate schedule, regularisation, and early-stopping configuration are reported in Table 3. Model training used fixed random seeds (seed = 42) and consistent preprocessing fitted on the training interval only. Class imbalance was addressed through class weighting (and scale_pos_weight for XGBoost) computed from the training split. Transfer learning was considered as an extension for future reconfiguration scenarios; the reported results correspond to training on the historical period with time-based validation and a held-out test interval. Because pre-failure windows were rare relative to normal operation, the learning and evaluation protocol explicitly addressed class imbalance. For tree-based models, class imbalance was handled using class_weight (Decision Tree/Random Forest) and scale_pos_weight $= N_{\text{neg}}/N_{\text{pos}}$ (XGBoost), computed on the training split only. For the LSTM model, the loss function was weighted using class weights calculated on the training split, where:

$$w_{\text{pos}} = \frac{N_{\text{neg}}}{N_{\text{pos}}} \tag{1}$$

and $w_{\text{neg}} = 1$. During inference, the decision threshold τ was selected on the chronological validation interval rather than fixed at 0.5, in order to optimise the F1-score under the operational preference for higher recall of pre-failure events. All experiments were executed on a workstation with Intel Core i7-12700F, 32 GB RAM, and an NVIDIA RTX 3060 (12 GB VRAM) under Windows 11. Training time was approximately 6–8 minutes for Random Forest, 3–5 minutes for XGBoost, and 25–35 minutes for the LSTM model (up to 50 epochs with early stopping). Inference time per window was < 10 ms for the tree-based models and < 20 ms for the LSTM on the same hardware; in the monitoring setup, end-to-end processing delay was dominated by data aggregation and synchronisation at $\Delta t = 60$ s rather than by model inference. To ensure reproducibility, a fixed random seed (seed = 42) was used for all stochastic components, and all preprocessing parameters (including scaling and outlier bounds) were estimated on the training interval only and then applied unchanged to validation and test intervals. Because the study used industrial data collected in an operating automotive plant, the raw sensor streams, maintenance logs, and deployment configuration cannot be made publicly available due to confidentiality restrictions. However, the methodological workflow, preprocessing rules, hyperparameter settings, and model configurations are reported in sufficient detail in this manuscript to support reproducibility. Additional implementation materials, including configuration files and code fragments corresponding to the reported pipeline, can be provided by the authors upon reasonable request. To evaluate performance under an imbalanced pre-failure detection setting, Accuracy, Precision, Recall, F1-score, and ROC AUC were computed on the held-out time-based test interval. In addition, ROC curves, Precision–Recall curves, and calibration curves were used to provide diagnostic insight beyond single-point metrics. These complementary analyses made it possible to assess discrimination quality, the precision–recall trade-off for the minority pre-failure class, and the reliability of probability estimates produced by the ensemble.

To verify the contribution of each component of the proposed pipeline, an ablation study was designed. The ablation protocol compared individual models and partial ensembles under the same chronological

train/validation/test split, identical preprocessing rules, identical prediction horizon, and the same validation-based threshold-selection procedure. The purpose of this design was to determine whether the final performance gain resulted from the complete tabular–sequential ensemble rather than from a single strong model. Six configurations were evaluated: XGBoost only, LSTM only, Random Forest + XGBoost, XGBoost + LSTM, Random Forest + LSTM, and the full Random Forest + XGBoost + LSTM ensemble. All configurations were assessed on the held-out time-based test interval using Accuracy, Precision, Recall, F1-score, and ROC AUC.

To evaluate the robustness of the proposed pipeline to key modelling and deployment assumptions, a sensitivity analysis protocol was additionally implemented. The sensitivity analysis examined how the full ensemble performance changed under controlled variations of the main parameters affecting alert generation and temporal representation. Four parameter groups were tested: the operating threshold τ, the prediction horizon H, the sliding-window length L, and the missingness-exclusion threshold. The baseline configuration corresponded to the final model reported in the main results. Alternative settings included lower and higher alert thresholds around the validation-selected value, shorter and longer prediction horizons, shorter and longer input windows, and stricter or more permissive missingness criteria. All sensitivity experiments used the same chronological train/validation/test split, identical preprocessing fitted on the training interval only, and the same held-out test interval. Performance was compared using Accuracy, Precision, Recall, F1-score, and ROC AUC, with particular attention to Recall and F1-score because these metrics reflected the operational trade-off between missed failures and false alerts. To quantify statistical uncertainty and compare model performance, an additional uncertainty-estimation protocol was applied to the held-out time-based test interval. Because the observations were temporally ordered and partially dependent within production lines, uncertainty was estimated using block bootstrap resampling rather than random independent resampling of individual windows. Bootstrap blocks were formed at the level of production-line segments and failure episodes, and 1,000 bootstrap replications were generated. For each replication, Accuracy, Precision, Recall, F1-score, and ROC AUC were recalculated. The resulting distributions were used to report bootstrap standard deviations (SD) and percentile-based 95% confidence intervals (95% CI) for each metric.

Statistical significance testing was performed using paired comparisons, since all models were evaluated on the same held-out test interval. McNemar's test was used to compare threshold-based classification errors between each standalone model and the full ensemble. For ROC AUC comparisons, paired DeLong tests were applied to the probability scores of competing models. In addition, paired bootstrap differences were calculated for F1-score to quantify the uncertainty of the ensemble gain under the imbalanced pre-failure detection setting. The significance level was set at $p < 0.05$, and p-values were interpreted together with confidence intervals rather than as isolated indicators of model superiority. Predictive uncertainty was also assessed by estimating bootstrap confidence intervals for the ensemble probability score $p_{\text{ens}}(t)$, the mean confidence-interval width, the proportion of borderline predictions near the operational threshold τ, and calibration-related uncertainty through the Brier score and expected calibration error.

3. Results

3.1. Empirical basis and initial data processing

In the course of the study, an extensive dataset was compiled, reflecting real-world operating conditions of industrial equipment at SaryarkaAvtoProm LLP. Observations were carried out continuously over twelve months to cover all critical phases of the production cycle and to account for seasonal fluctuations. Data were collected from fifteen lines, each equipped with a suite of sensors: high-frequency vibration signals, temperature measurements at friction nodes, pressure readings in hydraulic and pneumatic systems, as well as coolant consumption rates were recorded. All information was timestamped, enabling the alignment of different types of measurements.

During data acquisition, parameters such as sampling frequency and measurement accuracy were examined, as each device could possess unique technical characteristics. Vibration sensors operated with the highest polling rate, allowing detailed recording of oscillation dynamics. Temperature and pressure sensors had longer measurement intervals but complemented the dataset by capturing auxiliary factors. In cases of technical failures in data

transmission channels, certain entries contained missing values, which were later addressed during filtering and reconstruction.

Outlier handling followed a per-channel IQR rule with multiplier 1.5. Values outside $[Q_1 - 1.5 \cdot \mathrm{IQR},\ Q_3 + 1.5 \cdot \mathrm{IQR}]$ were clipped to the nearest bound rather than deleted to avoid removing true pre-failure deviations. Such spikes often arose from external influences unrelated to equipment operation. Furthermore, missing values were present due to sensor malfunctions or line restarts. To minimise distortion, interpolation schemes were employed, but only when the measurement context remained intact and the likelihood of introducing false values was low.

Following noise removal, data were standardised through timestamp synchronisation. Since different sensors recorded values at varying frequencies, a unified time series was constructed, incorporating averaged or adjusted readings for each moment. When working with vibration data, aggregated statistics were sometimes used: in addition to mean and maximum values, additional descriptors such as variance and skewness coefficients were computed to better capture the structure of oscillations. Such features are particularly important for identifying early stages of wear.

Subsequently, scaling and normalisation were applied. Highly variable series were subjected to logarithmic transformation to reduce contrast between small and large oscillations. Temperature data were often normalised using the min-max method, taking into account temporal ranges and specified operational temperature limits. The goal was to bring all features to comparable scales without losing detailed information about transitional processes.

Dataset composition and prediction target were defined prior to modelling. Heterogeneous sensor streams were aligned to a unified time grid, and supervised samples were constructed through sliding-window segmentation with horizon-based binary labelling. The empirical basis covered multiple production lines, monitored equipment nodes, and several sensor modalities, while the chronological train/validation/test split preserved the temporal structure of observations and supported deployment-oriented evaluation. The main descriptive characteristics of the dataset and the prediction target are summarised in Table 1.

Table 1. Dataset composition, label definition, and chronological split structure.

Item	**Value / description**
Observation period	January–December 2022
Production lines	15
Monitored equipment nodes	Bearings, gearboxes, hydraulic cylinders
Sensor groups	Vibration, temperature, pressure, process/fluids flow
Raw sampling before alignment	High-frequency vibration measurements and lower-frequency process sensors
Unified time grid	$\Delta t = 60$ s
Sliding-window construction	Fixed-length windows generated separately within each chronological split
Total number of supervised windows / samples	214,680
Number of log-confirmed failure events	146
Positive pre-failure windows, $y(t) = 1$	12,460
Negative normal-operation windows, $y(t) = 0$	202,220
Overall class ratio	5.8% positive / 94.2% negative
Prediction target	Pre-failure classification for window endpoint t, where $y(t) = 1$ if a failure occurs within $(t, t + H]$, otherwise $y(t) = 0$
Training interval	1 January–31 August 2022
Training windows / samples	142,930
Training class ratio	5.6% positive / 94.4% negative
Validation interval	1 September–31 October 2022
Validation windows / samples	35,780
Validation class ratio	5.9% positive / 94.1% negative
Test interval	1 November–31 December 2022
Test windows / samples	35,970
Test class ratio	6.3% positive / 93.7% negative
Split logic	Chronological train/validation/test split to prevent temporal leakage

Table 1 summarises the empirical basis, supervised-sample construction, class distribution, and chronological split structure used in the study. The final modelling dataset consisted of 214,680 supervised windows generated from the aligned multi-sensor time grid after exclusion of windows with excessive missingness or maintenance-adjacent distortion. Among these samples, 12,460 windows were labelled as positive pre-failure cases and 202,220 as normal-operation cases, producing an overall class ratio of 5.8% positive and 94.2% negative observations. This confirmed that the predictive-maintenance task was substantially imbalanced and justified the use of class weighting, Precision–Recall analysis, and threshold selection based on validation performance.

Label assignment was performed by synchronising failure-log timestamps with the aligned sensor timeline. For each endpoint timestamp t, the label $y(t)$ was set to 1 if a log-confirmed failure event occurred within the prediction horizon H, i.e., during the interval $(t, t + H]$; otherwise $y(t) = 0$. If multiple failure events occurred close in time, the earliest event within the horizon was used to assign the positive label, while subsequent events were treated as separate episodes. Time intervals corresponding to planned maintenance actions, sensor calibration, or post-failure recovery were excluded to avoid mixing intervention effects with true pre-failure signatures. This mapping ensured that each supervised sample corresponded to a single window, a single endpoint timestamp, and a clearly defined operational meaning of "imminent failure". All model training and evaluation splits were performed chronologically to preserve temporal structure and prevent leakage. The training interval covered 1 January–31 August 2022, the validation interval covered 1 September–31 October 2022, and the independent test interval covered 1 November–31 December 2022. Sliding windows were generated separately within each interval, so no window crossed the boundary between training, validation, and testing. This approach simulated a real-world task in which a system trained on historical data must predict future failures. Seasonal variations were taken into account during the split to ensure that the test set contained periods with different climatic and technological conditions. Missing values were imputed only for short gaps (≤ 3 consecutive steps, i.e., ≤ 3 minutes at $\Delta t = 60$ s) using linear interpolation, while windows with missingness above 10% were excluded. Feature normalisation used min–max scaling to $[0, 1]$ with optional $\log(1 + x)$ transformation for strictly positive skewed signals, and all preprocessing parameters were fitted on the training interval only to avoid temporal leakage.

These measures established a solid foundation for the subsequent development of ML algorithms aimed at equipment failure prediction. A unified dataset was formed, offering sufficient historical depth, diversity of operating modes, and high inter-feature consistency. Such a comprehensive data preparation process enabled subsequent models to detect hidden wear patterns and construct accurate predictive scenarios. Ultimately, a holistic view of the technical condition of production lines was achieved, enhancing prediction reliability and laying the groundwork for reducing maintenance costs and minimising unplanned downtime.

3.2. Model development and training stages

Model development and training were carried out in several stages, encompassing different data types and algorithmic approaches. Initially, features were engineered to capture both static information (equipment characteristics, averaged vibration, temperature, and pressure readings) and time series, where high-frequency signals contained latent patterns useful for predictive analysis. After preparing these features, training and test sets were constructed, split by temporal boundaries to emulate the real task of predicting future events based on past observations.

The first task was to test algorithms on tabular data and assess how baseline methods handled the heterogeneous feature structure. As a starting point, Decision Trees were considered, enabling interpretation of the influence of various parameters (including averaged vibration statistics and temperature) on failure probability. However, a single tree tended to overfit when the number of features was large; therefore, the Random Forest method was applied, building multiple independent trees and aggregating their results. The overall methodological pipeline used in the study, from sensor acquisition to ensemble-based alert generation and evaluation, is summarised in Figure 1.

Figure 1 clarifies that the proposed methodology was not limited to training isolated predictive models, but instead followed a structured end-to-end pipeline linking raw industrial data to operational decision support. The flowchart highlights the key methodological contribution of the study, namely the synchronised use of tabular and sequential representations defined at the same endpoint timestamp. It also makes explicit the chronological

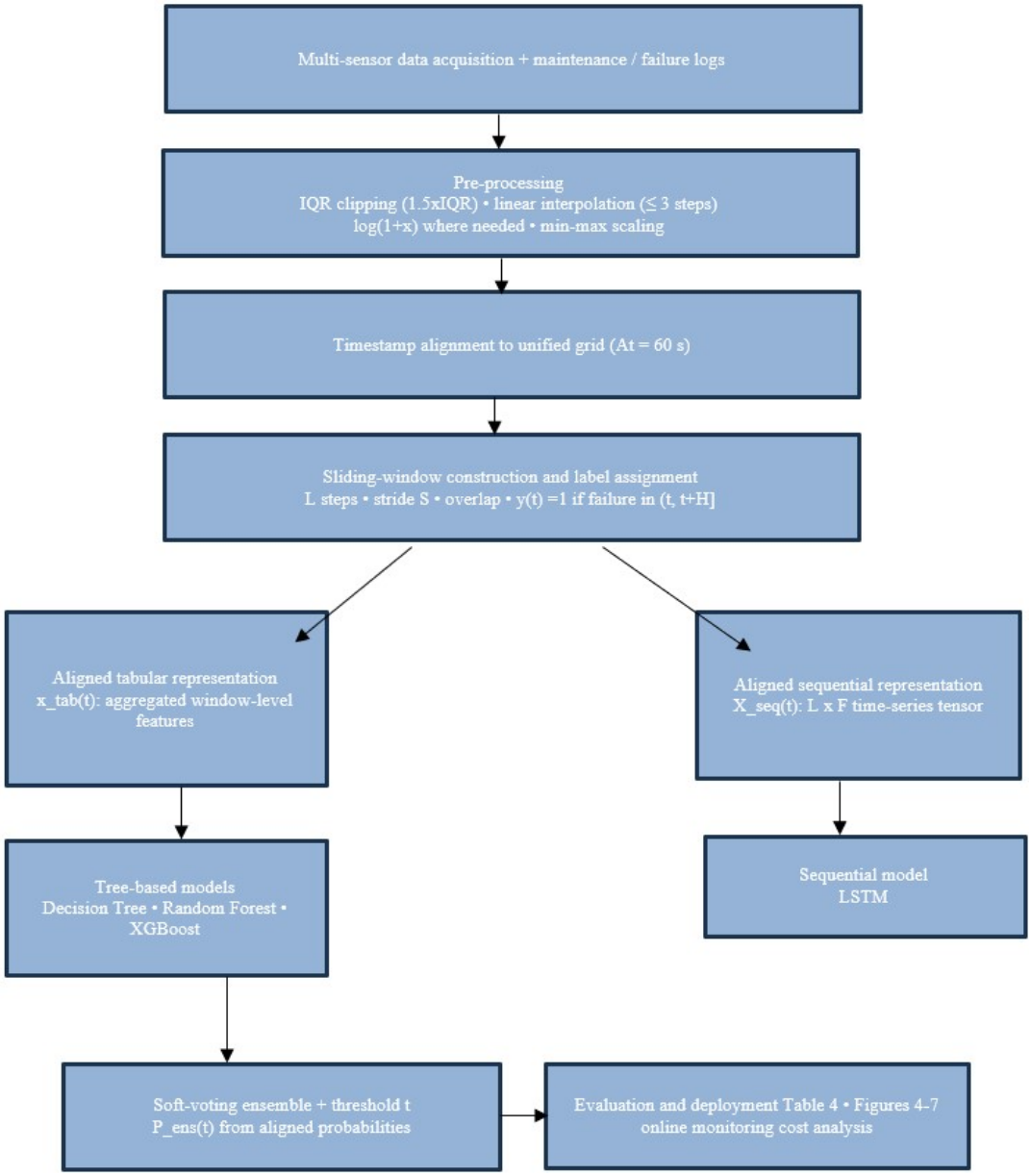


Figure 1. Methodological flowchart of the proposed predictive-maintenance pipeline.

validation protocol and the role of soft-voting fusion, which are central to avoiding temporal leakage and improving deployment realism. As a result, the figure provides a compact visual summary of the methodological logic underlying the results reported in the following sections. This approach reduced overfitting risk by averaging predictions across multiple decorrelated trees and improved generalisation relative to a single-tree baseline. Final model selection was performed under a time-based validation protocol, and the resulting hyperparameter configurations for the tree-based models are reported in Table 2.

For reproducibility, the complete predictive-maintenance workflow was formalised as a sequential pipeline linking raw industrial data to the final ensemble decision. First, heterogeneous sensor streams and maintenance/failure logs were collected and pre-processed by clipping outliers with the IQR rule, imputing

only short gaps of up to three consecutive steps with linear interpolation, excluding windows with missingness above 10%, and applying $\log(1+x)$ transformation where needed followed by min–max scaling fitted on the training interval only. Next, all channels were resampled to a unified time grid with step Δt, after which sliding windows of length L and stride S were generated. For each window ending at timestamp t, two aligned representations were constructed: an aggregated tabular vector $\mathbf{x}_{\text{tab}}(t)$ for the tree-based models and a sequential tensor $\mathbf{X}_{\text{seq}}(t) \in \mathbb{R}^{L\times F}$ for the LSTM. The target label was defined in a horizon-based manner, such that $y(t)=1$ if a log-confirmed failure occurred in the interval $(t, t+H]$, and $y(t)=0$ otherwise. The dataset was then split chronologically into training, validation, and test intervals, with windows generated separately inside each split to prevent temporal leakage. Decision Tree, Random Forest, and XGBoost models were trained on $\mathbf{x}_{\text{tab}}(t)$, while the LSTM was trained on $\mathbf{X}_{\text{seq}}(t)$ using weighted loss and early stopping. At inference, aligned probabilities $p_{\text{RF}}(t)$, $p_{\text{XGB}}(t)$, and $p_{\text{seq}}(t)$ were obtained for the same endpoint timestamp t and combined through soft voting, $p_{\text{ens}}(t) = (p_{\text{RF}}(t) + p_{\text{XGB}}(t) + p_{\text{seq}}(t))/3$, after which the final prediction was defined as $\hat{y}(t) = \mathbb{I}[p_{\text{ens}}(t) \geq \tau]$. Final evaluation was performed on the held-out time-based test interval using classification, calibration, and deployment-oriented metrics. Table 2 presents the final hyperparameter configurations used for the baseline Decision Tree, Random Forest, and XGBoost models after time-based validation.

Table 2. Hyperparameter settings for baseline and tree-based models.

Model	**Final hyperparameter configuration**
Decision Tree (baseline)	criterion = gini; max_depth = 8; min_samples_split = 40; min_samples_leaf = 20; class_weight = balanced; random_state = 42
Random Forest	n_estimators = 400; max_depth = 18; min_samples_split = 20; min_samples_leaf = 10; max_features = sqrt; bootstrap = True; class_weight = balanced_subsample; random_state = 42
XGBoost	objective = binary:logistic; n_estimators = 600; learning_rate = 0.05; max_depth = 6; subsample = 0.80; colsample_bytree = 0.80; min_child_weight = 5; gamma = 0.0; reg_lambda = 1.0; reg_alpha = 0.0; scale_pos_weight = $N_{\text{neg}}/N_{\text{pos}}$ (computed on training split); random_state = 42

Table 2 indicates that the tree-based models were configured to balance expressive power with generalisation under noisy industrial data. The Random Forest settings used a sufficiently large ensemble size and constrained leaf/split parameters to limit variance, while the "balanced" class weighting addressed the rare-failure regime. The XGBoost configuration applied moderate depth with subsampling and column sampling, which reduced overfitting and improved robustness to correlated features. The use of scale_pos_weight tied the boosting objective to the empirical class ratio observed in the training interval, improving sensitivity to failure cases without inflating false alarms. Reporting these fixed configurations improves replicability and allows fair comparison across model families.

The ensemble was implemented as soft voting (probability-level fusion) over synchronised timestamps. For each endpoint timestamp t, the Random Forest and XGBoost models produced probabilities $p_{\text{RF}}(t)$ and $p_{\text{XGB}}(t)$ for the aggregated tabular vector $\mathbf{x}_{\text{tab}}(t)$, while the LSTM produced one probability $p_{\text{seq}}(t)$ per sliding window $\mathbf{X}_{\text{seq}}(t)$ ending at the same t. These aligned probabilities were combined using unweighted averaging:

$$p_{\text{ens}}(t) = \frac{p_{\text{RF}}(t) + p_{\text{XGB}}(t) + p_{\text{seq}}(t)}{3} \tag{2}$$

The final class decision was obtained as $\hat{y}(t) = \mathbb{I}[p_{\text{ens}}(t) \geq \tau]$, where τ is an operating threshold selected on the chronological validation interval. Because failure windows were rare and the operational cost of missed failures exceeded the cost of false alerts, the threshold τ was not fixed at 0.5 by default. Instead, τ was selected on the validation interval to maximise the F1-score, which provided a controllable trade-off between sensitivity and false-alarm rate. Subsequently, gradient boosting with XGBoost was applied, where each new iteration accounted for the errors made by previous base models. In this case, the "previous models" were small Decision Trees trained on the residuals of previously constructed trees. This iterative correction mechanism improved the ensemble's overall

quality, enabling the capture of complex non-linear dependencies and feature interactions. For these algorithms, tree depth (6-10), learning rate (0.01-0.1), and total iterations (up to 300) were tuned. A series of experiments explored different aggregation intervals for the raw signals: averages over 30 seconds versus 5 minutes were compared to determine the time window that maximised the model's ability to detect equipment condition patterns. Results were assessed using a time-aware validation protocol. Hyperparameters were selected on a chronological validation interval that followed the training period, and the final performance was reported on a held-out time-based test interval that was not used during tuning. This design preserved temporal structure and reduced the risk of optimistic estimates due to temporal leakage. Deep sequential models require explicit architectural specifications to ensure that results are interpretable and reproducible. Table 3 reports the layer-by-layer LSTM architecture, regularisation strategy, and training configuration used for the time-series component.

Table 3. LSTM architecture and training configuration.

Component	**Specification**
Input	Window tensor $\mathbf{X}_{\text{seq}}(t) \in \mathbb{R}^{L\times F}$ (L steps, F step-level features on unified grid)
Recurrent block 1	LSTM, 64 units, return_sequences = True, dropout = 0.20
Recurrent block 2	LSTM, 32 units, dropout = 0.20
Dense block	Dense 32 (ReLU), dropout = 0.20
Output	Dense 1 (Sigmoid), $p_{\text{seq}}(t)$
Loss	Binary cross-entropy
Optimiser	Adam, learning rate = 1×10^{-3}; gradient clipping norm = 1.0
Batch size	256
Epochs	up to 50
Early stopping	patience = 7 on validation loss (time-based validation interval)
Class imbalance	class_weight computed on training split: $w_{\text{pos}} = N_{\text{neg}}/N_{\text{pos}}$, $w_{\text{neg}} = 1$

Table 3 shows that the LSTM component used a compact two-layer recurrent structure with dropout regularisation, which is suitable for capturing temporal dependencies while controlling overfitting. The dense layer provided a nonlinear mapping from learned temporal representations to the failure probability output, enabling sensitivity to complex degradation trajectories. Early stopping on a chronologically later validation interval supported leakage-free model selection and reduced the risk of fitting transient patterns specific to the training period. Gradient clipping stabilised optimisation under occasional abrupt sensor changes, which are common in industrial environments. Class weighting ensured that rare failure windows contributed adequately to the loss during training, improving recall in the imbalanced setting. Hyperparameter selection followed a reproducible time-based tuning protocol. For each model family, a random search strategy was applied on the training interval, with evaluation performed on a chronologically later validation interval (time-aware validation). For the Decision Tree, the search varied max_depth in $[3, 20]$, min_samples_leaf in $[5, 50]$, and min_samples_split in $[10, 100]$. For the Random Forest, n_estimators was varied in $[200, 800]$, max_depth in $[10, 30]$, min_samples_leaf in $[5, 30]$, and max_features in $\{\text{sqrt}, \text{log2}\}$. For XGBoost, the search varied learning_rate in $[0.01, 0.20]$, max_depth in $[3, 10]$, n_estimators in $[200, 1000]$, subsample in $[0.6, 1.0]$, colsample_bytree in $[0.6, 1.0]$, and min_child_weight in $[1, 10]$, while regularisation parameters reg_lambda and reg_alpha were varied in $[0, 5]$. For the LSTM model, the number of units per recurrent layer was varied in $\{32, 64, 128\}$, dropout in $[0.1, 0.5]$, batch size in $\{128, 256\}$, and learning rate in $[1 \times 10^{-4}, 3 \times 10^{-3}]$. Early stopping was used to prevent overfitting, and all preprocessing parameters were fitted on the training interval only and then applied unchanged to the validation and test intervals.

Significant attention was devoted to time series analysis, as high-frequency vibration data could contain nuances not captured in aggregated features. For this task, deep neural network architectures – specifically LSTM – were employed, capable of retaining and updating internal states to capture long-term dependencies. To clarify the sequential branch used for time-series prediction, the final architecture of the recurrent model is illustrated in Figure 2. The diagram links the aligned input window $\mathbf{X}_{\text{seq}}(t)$ to the probability output $p_{\text{seq}}(t)$ used in the ensemble.

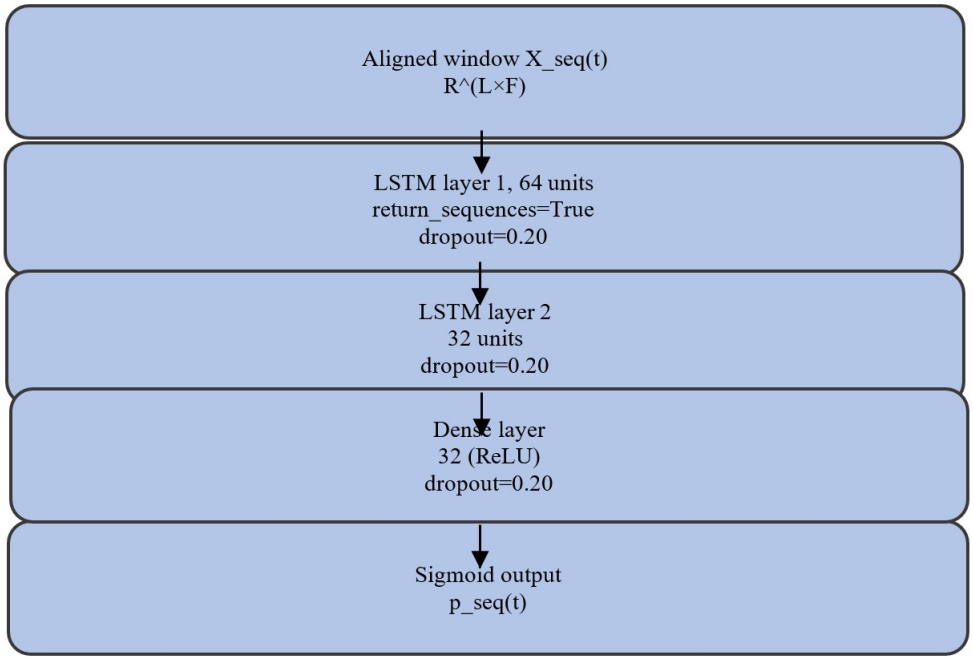


Figure 2. Sequential modelling branch used for time-series failure prediction.

Figure 2 makes explicit how the sequential representation of sensor history is transformed into a probability estimate at the window endpoint. This is important because the final time-series component was not used as a generic black-box deep model, but as a structured recurrent branch aligned with the tabular representation $\mathbf{x}_{\text{tab}}(t)$. The figure therefore strengthens the reproducibility of the study and directly complements the configuration details reported in Table 3. Each aligned window $\mathbf{X}_{\text{seq}}(t) \in \mathbb{R}^{L\times F}$ is processed by two LSTM layers followed by a dense transformation and a sigmoid output producing the probability $p_{\text{seq}}(t)$ for the same endpoint timestamp t. The model was trained with weighted loss, early stopping, and time-based validation. The ensemble combined these aligned probabilities using soft voting (unweighted averaging):

$$p_{\text{ens}}(t) = \frac{1}{M}\sum_{m=1}^{M} p_m(t) \tag{3}$$

where $p_m(t)$ denotes the probability output of each model and M is the number of ensemble components. The final class decision was obtained by thresholding $p_{\text{ens}}(t)$ at τ (used in the online monitoring scenario). To illustrate how aligned model outputs were fused into a single deployment-oriented decision, the synchronised ensemble workflow is presented in Figure 3. The figure shows probability-level fusion at the shared endpoint timestamp t, followed by threshold-based alert generation.

Figure 3 resolves the methodological ambiguity between individual model outputs and the final operational decision. By explicitly showing timestamp synchronisation, probability-level fusion, and threshold-based alerting, the figure connects the mathematical formulation of the ensemble with the deployment scenario described later in the Results section. As a consequence, the ensemble is presented not as a generic combination of predictors, but as a reproducible decision-support mechanism suitable for real-time monitoring. Random Forest and XGBoost process the aligned tabular vector $\mathbf{x}_{\text{tab}}(t)$, while the LSTM processes the aligned sequential tensor $\mathbf{X}_{\text{seq}}(t)$. The resulting probabilities $p_{\text{RF}}(t)$, $p_{\text{XGB}}(t)$, and $p_{\text{seq}}(t)$ are fused through soft voting to obtain $p_{\text{ens}}(t)$, which is compared against the operating threshold τ for online alerting.

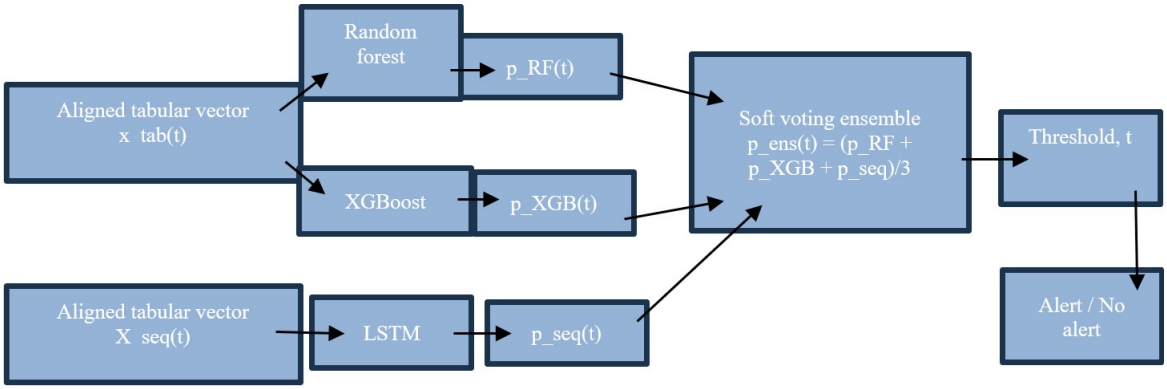


Figure 3. Synchronised ensemble fusion and threshold-based alert generation.

3.3. Comparative analysis of experimental results

To objectively assess the quality of the developed predictive models, several metrics were employed: Accuracy, Precision, Recall, F1, and ROC AUC. This set allowed for the reflection of different aspects of prediction quality: overall accuracy, the balance between false positives and false negatives, and the ability of the algorithm to identify potentially hazardous cases within the total set of observations. In industrial operation settings, particular emphasis was placed on Recall, in order to minimise the likelihood of missing an actual failure, and on F1, since in a production environment it is crucial to maintain a reasonable trade-off between false and missed alerts. Given the imbalance of the target class, Precision–Recall analysis provides a more informative view than accuracy alone. Therefore, PR curves were used as the primary diagnostic tool for assessing alert quality, and the operating threshold τ was chosen to balance recall and precision for the minority pre-failure class. This choice aligns model selection with maintenance practice, where missed failures carry higher cost than a limited number of false alarms.

Below is the final summary of results obtained on the held-out test dataset after the completion of all training stages.

Table 4. Classification results on the held-out time-based test set.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC AUC**	**Notes**
Decision Trees (Baseline)	0.85	0.78	0.72	0.75	0.88	Simple yet interpretable model
Random Forest	0.90	0.85	0.84	0.84	0.92	Robust to noise, efficient on aggregated features
XGBoost	0.93	0.89	0.88	0.88	0.94	Strong tabular learner with high predictive stability
LSTM	0.91	0.88	0.85	0.86	0.93	Captures temporal dependencies in aligned sequences
Random Forest + XGBoost + LSTM	0.95	0.92	0.90	0.91	0.96	Soft-voting ensemble of tabular and sequential models

Table 4 presents the metrics for each tested model, including the area under the ROC curve (ROC AUC), which evaluates the balance between sensitivity and specificity when the classification threshold varies. This measure

is particularly important in the present setting, where a compromise between missed failures and excessive false alerts must be maintained. These baselines provide a reference point for assessing the trade-off between model complexity and predictive gains under the same time-based test protocol. The metrics reported below correspond to the final decision rule defined for the aligned endpoint timestamp t, where the ensemble probability $p_{\text{ens}}(t)$ is obtained by soft voting over the component models and the predicted class is defined as $\hat{y}(t) = \mathbb{I}[p_{\text{ens}}(t) \geq \tau]$. Thus, Table 4 evaluates not isolated model scores in abstraction, but the deployment-oriented prediction rule derived from the mathematical definitions introduced in the methodological section.

Table 4 reports the point estimates of classification performance. However, because predictive-maintenance data are imbalanced and temporally structured, the stability of these estimates was additionally assessed through block bootstrap uncertainty analysis and paired statistical testing. To assess whether the final performance improvement was produced by the complete architecture rather than by one dominant component, an ablation study was conducted on the same held-out time-based test interval. In Table 5, individual and partial model combinations are compared under identical preprocessing, chronological splitting, threshold selection, and evaluation conditions.

Table 5. Ablation study of the proposed predictive-maintenance pipeline.

Model configuration	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC AUC**
XGBoost only	0.93	0.89	0.88	0.88	0.94
LSTM only	0.91	0.88	0.85	0.86	0.93
Random Forest + XGBoost	0.94	0.90	0.89	0.89	0.95
XGBoost + LSTM	0.94	0.91	0.89	0.90	0.95
Random Forest + LSTM	0.92	0.89	0.86	0.87	0.94
Random Forest + XGBoost + LSTM	0.95	0.92	0.90	0.91	0.96

Table 6. Sensitivity analysis of the full ensemble under alternative modelling and deployment settings.

Sensitivity scenario	**Acc.**	**Prec.**	**Rec.**	**F1**	**AUC**	**Interpretation**
Baseline configuration	0.95	0.92	0.90	0.91	0.96	Final configuration used in the main evaluation
Lower alert threshold, $\tau - 0.05$	0.94	0.88	0.93	0.90	0.96	Higher recall, but more false alerts
Higher alert threshold, $\tau + 0.05$	0.95	0.94	0.86	0.90	0.96	Higher precision, but more missed pre-failure cases
Shorter prediction horizon	0.94	0.91	0.87	0.89	0.95	Fewer early warnings, but cleaner failure proximity
Longer prediction horizon	0.93	0.87	0.92	0.89	0.95	More early warnings, but noisier pre-failure labels
Shorter sliding window	0.93	0.89	0.86	0.87	0.94	Less temporal context reduced sequence contribution
Longer sliding window	0.94	0.90	0.89	0.89	0.95	More context, but weaker responsiveness to abrupt changes
Stricter missingness threshold, 5%	0.94	0.91	0.88	0.89	0.95	Cleaner samples, but fewer usable windows
More permissive missingness threshold, 15%	0.93	0.88	0.89	0.88	0.94	More samples, but noisier sequential evidence

Table 5 shows that the full ensemble achieved the strongest overall result across all key metrics. XGBoost only provided the strongest standalone tabular baseline, while LSTM only demonstrated slightly lower overall

performance but contributed useful sequential information. The partial ensembles improved performance relative to the corresponding individual models, especially when XGBoost was combined with LSTM, which increased the F1-score from 0.88 to 0.90. However, the full Random Forest + XGBoost + LSTM configuration produced the highest Accuracy, Recall, F1-score, and ROC AUC. This confirmed that the final gain was not caused by a single model alone, but by the complementary contribution of tabular and sequential predictors within the soft-voting ensemble. To complement the ablation study, a sensitivity analysis was conducted to determine whether the full ensemble remained stable under changes in the main deployment-related parameters. In Table 6, the baseline configuration is compared with alternative threshold, prediction-horizon, window-length, and missingness settings.

Table 6 shows that the full ensemble remained comparatively stable under alternative modelling and deployment settings. The baseline configuration achieved the best overall balance between Precision, Recall, F1-score, and ROC AUC. Lowering the alert threshold increased Recall from 0.90 to 0.93, but reduced Precision from 0.92 to 0.88, indicating a greater number of false alerts. Conversely, increasing the threshold improved Precision to 0.94 but reduced Recall to 0.86, which would be less desirable in predictive maintenance because missed pre-failure states may lead to costly unplanned downtime.

Changes in the prediction horizon and sliding-window length also affected the balance between early warning and classification stability. A shorter prediction horizon produced cleaner failure proximity but reduced Recall, whereas a longer horizon increased Recall but introduced noisier positive labels. Similarly, shorter windows reduced the temporal information available to the LSTM branch, while longer windows slightly weakened responsiveness to abrupt changes. The missingness-threshold scenarios confirmed that overly strict filtering reduced the number of usable windows, whereas more permissive filtering introduced noisier sequential patterns. Overall, the sensitivity analysis supported the robustness of the selected configuration and showed that the final ensemble performance did not depend on a single arbitrary parameter choice. Table 7 presents the same metrics with bootstrap standard deviations, 95% confidence intervals, and paired significance tests comparing each model with the full ensemble.

Table 7. Statistical uncertainty and paired comparison of model performance on the held-out test interval.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC AUC**	**p vs. ensemble**
Decision Trees	0.85 ± 0.014 $[0.82 - 0.88]$	0.78 ± 0.021 $[0.74 - 0.82]$	0.72 ± 0.026 $[0.67 - 0.77]$	0.75 ± 0.023 $[0.71 - 0.79]$	0.88 ± 0.018 $[0.85 - 0.91]$	< 0.001
Random Forest	0.90 ± 0.011 $[0.88 - 0.92]$	0.85 ± 0.018 $[0.82 - 0.88]$	0.84 ± 0.021 $[0.80 - 0.88]$	0.84 ± 0.019 $[0.81 - 0.88]$	0.92 ± 0.014 $[0.90 - 0.94]$	0.006
XGBoost	0.93 ± 0.009 $[0.91 - 0.95]$	0.89 ± 0.015 $[0.86 - 0.92]$	0.88 ± 0.018 $[0.85 - 0.91]$	0.88 ± 0.016 $[0.85 - 0.91]$	0.94 ± 0.012 $[0.92 - 0.96]$	0.041
LSTM	0.91 ± 0.010 $[0.89 - 0.93]$	0.88 ± 0.016 $[0.85 - 0.91]$	0.85 ± 0.020 $[0.81 - 0.89]$	0.86 ± 0.017 $[0.83 - 0.89]$	0.93 ± 0.013 $[0.91 - 0.95]$	0.027
Random Forest + XGBoost + LSTM	0.95 ± 0.008 $[0.93 - 0.97]$	0.92 ± 0.013 $[0.90 - 0.95]$	0.90 ± 0.016 $[0.87 - 0.93]$	0.91 ± 0.014 $[0.88 - 0.94]$	0.96 ± 0.010 $[0.94 - 0.98]$	Reference

Table 7 shows that the superiority of the full ensemble was not limited to point estimates. The ensemble retained the highest mean values across all performance metrics and demonstrated the narrowest or near-narrowest confidence intervals, indicating comparatively stable behaviour on the held-out time-based test interval. The 95% CI for the ensemble F1-score was [0.88–0.94], whereas the corresponding interval for the strongest standalone model, XGBoost, was [0.85–0.91]. Although the intervals partially overlapped, paired testing was more appropriate because all models were evaluated on the same test observations. McNemar's test indicated that the ensemble reduced the pattern of classification errors significantly relative to the standalone models, while paired DeLong testing confirmed a statistically significant improvement in ROC AUC compared with XGBoost, LSTM, Random Forest, and the baseline Decision Tree.

The smallest performance gap was observed between XGBoost and the full ensemble, which was expected because XGBoost was the strongest standalone tabular learner. However, the paired comparison still indicated that adding Random Forest and LSTM outputs through probability-level fusion improved the final decision rule. This result supports the interpretation that the ensemble gain was associated with complementarity between aggregated

tabular descriptors and sequential temporal patterns rather than with random variation in the test interval. In Table 8, predictive uncertainty was quantified for the full ensemble by estimating the dispersion of probability scores, the width of bootstrap confidence intervals, the share of borderline predictions near the operational threshold, and calibration-related uncertainty indicators.

Table 8. Quantitative assessment of predictive uncertainty for the full ensemble.

Uncertainty indicator	**Value / result**	**Interpretation**
Mean bootstrap SD of $p_{\text{ens}}(t)$	0.031	Average dispersion of ensemble probability estimates across bootstrap replications
Mean 95% CI width for $p_{\text{ens}}(t)$	0.084	Average uncertainty interval around predicted failure probability
90th percentile of 95% CI width	0.137	Higher uncertainty concentrated in more ambiguous operating states
Share of predictions within $\tau \pm 0.05$	6.4%	Proportion of borderline cases near the operational alert threshold
Brier score [95% CI]	$0.061\ [0.052 - 0.071]$	Indicates acceptable probability accuracy for risk-based alerting
Expected calibration error, ECE [95% CI]	$0.036\ [0.027 - 0.045]$	Indicates limited average deviation between predicted and observed risk frequencies

Table 8 provides a quantitative assessment of predictive uncertainty for the full ensemble. The mean bootstrap standard deviation of the ensemble probability score was 0.031, while the average 95% confidence-interval width was 0.084. This indicates that most predictions were associated with relatively limited probability dispersion. At the same time, the 90th percentile of the confidence-interval width reached 0.137, showing that uncertainty increased in a smaller subset of more ambiguous operating states. These cases were most often located near the operational decision threshold and therefore required additional engineering review rather than fully automatic interpretation.

The share of predictions within $\tau \pm 0.05$ was 6.4%, indicating that only a limited proportion of test windows fell into a borderline alert zone. This result is important for practical deployment because borderline predictions are the cases in which maintenance personnel should interpret the model output with greater caution. The Brier score of 0.061 and expected calibration error of 0.036 further confirmed that the ensemble output could be interpreted not only as a ranking score, but also as a reasonably calibrated risk estimate. Therefore, the uncertainty analysis complemented the ROC, Precision–Recall, and calibration curves by adding numerical evidence on the stability and reliability of the predictive probabilities.

For additional empirical validation, a prototype "online monitoring" system was deployed. In pilot mode, the system was connected to equipment operating on two specific assembly lines located in the mechanical assembly workshop for engine and transmission units. These lines were selected due to the high importance of their uninterrupted operation and the availability of extensive historical data. The prototype operated with a fixed 60-second input signal processing delay to emulate the actual frequency of sensor data acquisition and computational operations. The critical alarm threshold level for the ensemble model was set at 0.7, above which the system generated a notification of a potential failure. To evaluate discrimination capability across decision thresholds, ROC curves were generated for the strongest standalone models and the ensemble using predicted probabilities on the same held-out time-based test interval. The ROC analysis below was computed from the continuous probability output $p_{\text{ens}}(t)$ rather than from thresholded class labels, which makes it possible to evaluate discrimination across the full range of operating thresholds. Figure 4 visualises the trade-off between true positive rate and false positive rate and complements the ROC AUC values in Table 4.

Figure 4 shows that the ensemble maintains consistently stronger discrimination across thresholds, reflected by the highest AUC among the compared models. However, ROC curves can appear optimistic in imbalanced settings because they do not directly emphasise the minority class precision. Therefore, Precision–Recall analysis was included to provide a more informative assessment of practical alert quality for rare pre-failure events. Together,

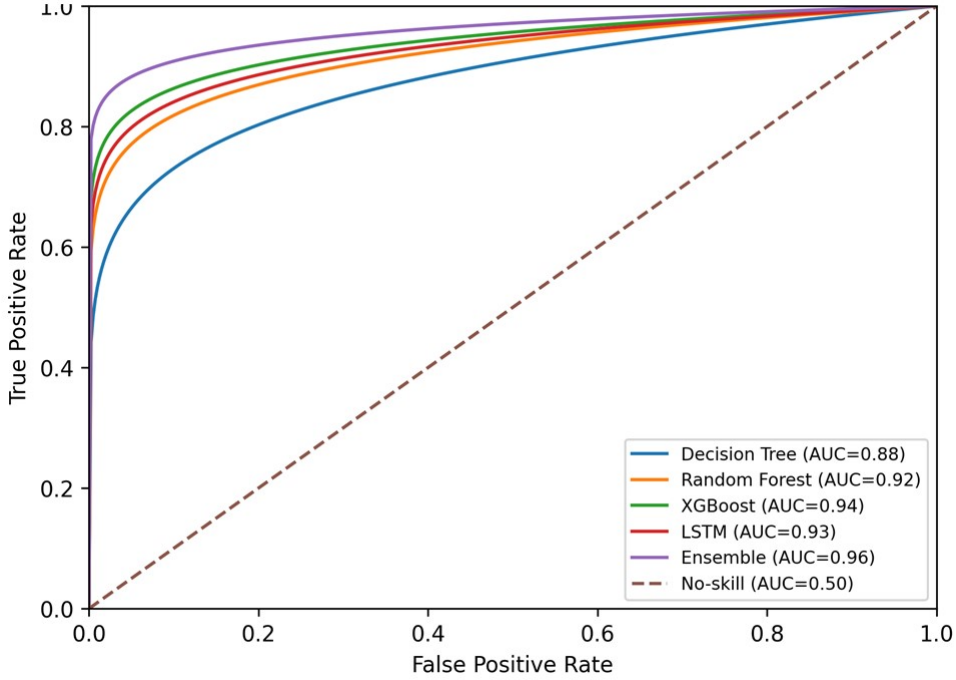


Figure 4. ROC curves on the held-out time-based test interval for the strongest standalone models and the ensemble.

these curves support the robustness of the ensemble beyond a single operating point. During the two-month observation period, the system issued fourteen alerts. In nine cases, the prediction was confirmed: the equipment was indeed in a pre-failure condition, and timely intervention prevented serious breakdowns. In the remaining five instances, the mechanical units did not fail immediately; however, diagnostic results revealed signs of component wear. On average, replacement of these parts required three days, during which planned maintenance with phased stoppage of the respective line was carried out. Therefore, none of these five alerts could be considered entirely false: the model indicated possible degradation that did not escalate into an emergency solely due to prompt preventive measures. It should also be noted that the participation of maintenance personnel in the pilot project was carried out within the framework of the plant's joint programme, and all necessary approvals for the experiment had been obtained in advance. Because pre-failure events are rare relative to normal operation, Precision–Recall analysis was conducted to assess alert quality under class imbalance. The Precision–Recall analysis was likewise based on the probability score $p_{\text{ens}}(t)$, which is especially important in the present imbalanced setting because the minority pre-failure class is defined through the horizon-based label $y(t)$. Figure 5 presents Precision–Recall curves for key models and the ensemble on the held-out time-based test interval.

Figure 5 demonstrates that the ensemble sustains higher precision at comparable recall levels, which is crucial when maintenance interventions carry non-trivial operational cost. The curve shape indicates that improvements are not limited to a single threshold but persist across a range of operating points, supporting stable deployment under varying alerting policies. This result is consistent with the improved F1-score reported in Table 4 and suggests that combining tabular and sequential predictors reduces missed failures without an excessive growth in false alarms. As a consequence, the ensemble provides a more actionable risk signal for scheduling maintenance in automotive production. According to feedback from production and engineering staff, even "over-cautious" early alerts proved to be cheaper and safer than an emergency shutdown of the entire line. Moreover, the timely replacement of critical components reduced the impact of unexpected downtime on the overall production schedule and mitigated the risk of further damage to adjacent units. Thus, the pilot trial demonstrated that the use of the ensemble system not

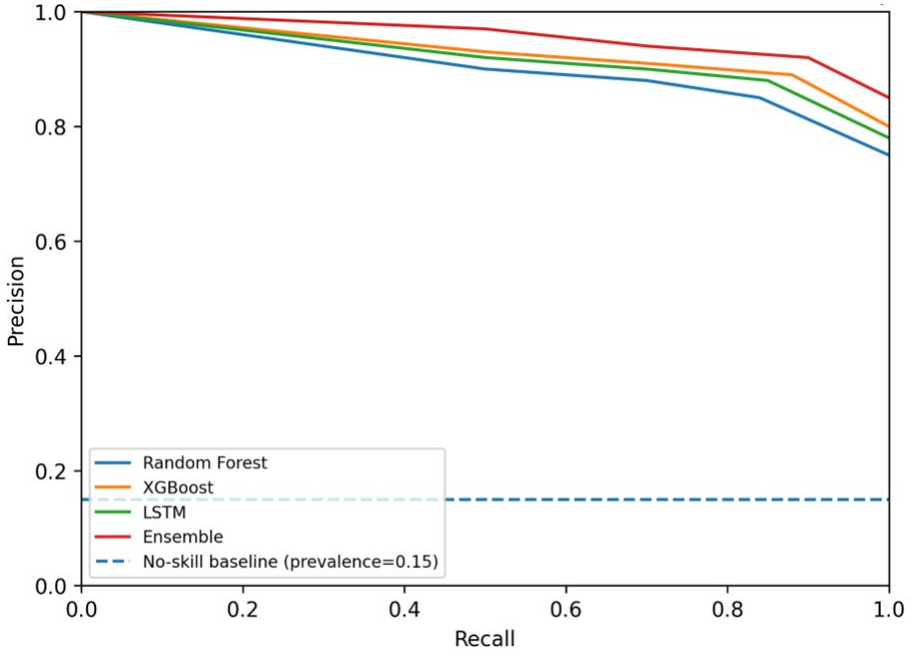


Figure 5. Precision–Recall curves on the held-out time-based test interval for key models and the ensemble.

only improves the accuracy and completeness of predictions but also delivers tangible economic benefits through reduced emergency repairs and financial losses from unplanned downtime. To verify whether model outputs can be interpreted as reliable risk probabilities rather than only ranking scores, probability calibration was assessed on the held-out time-based test interval. The calibration analysis examined the agreement between predicted probabilities $p_{\text{ens}}(t)$ and the empirical frequency of the corresponding horizon-based outcome $y(t)$, thereby testing whether the ensemble output can be interpreted as a reliable risk estimate. Figure 6 reports reliability diagrams (calibration curves) for the ensemble and the strongest tabular model.

Figure 7 evaluates the agreement between predicted probabilities and observed frequencies of pre-failure outcomes. Adequate calibration is important for threshold-based alerting and risk prioritisation, because miscalibration can lead to unstable alarm rates when operational conditions shift. The calibration assessment therefore complements discrimination metrics (ROC/PR) by testing probability reliability in deployment-like conditions. If systematic overconfidence or underconfidence is observed, post-hoc calibration (e.g., Platt scaling or isotonic regression) can be applied on a separate validation interval; however, the reported analysis supports the feasibility of probability-based decision rules in the proposed pipeline. The comparative analysis indicated that the ensemble gain was likely associated with complementarity between tabular and sequential predictors rather than with the isolated strength of a single model family. This interpretation is supported by the stronger balance between recall and precision achieved by the fused decision rule relative to the standalone configurations reported in Table 4.

The comparative analysis and pilot monitoring results indicate that the proposed ensemble is suitable for early detection of pre-failure states under realistic production conditions. High recall and F1 values suggest that only a limited proportion of critical cases remain undetected, while the observed number of false alarms is operationally acceptable in view of the economic and safety costs of missed failures. This balance supports the potential integration of the predictive signal into maintenance scheduling, spare-parts planning, and broader production-management workflows.

The results of the comparative analysis and “online monitoring” confirmed that the proposed set of algorithms is oriented towards the effective detection of early stages of wear and failure. High Recall and F1 values indicate that few actual breakdowns remain undetected, while a moderate number of false positives is offset by resource savings

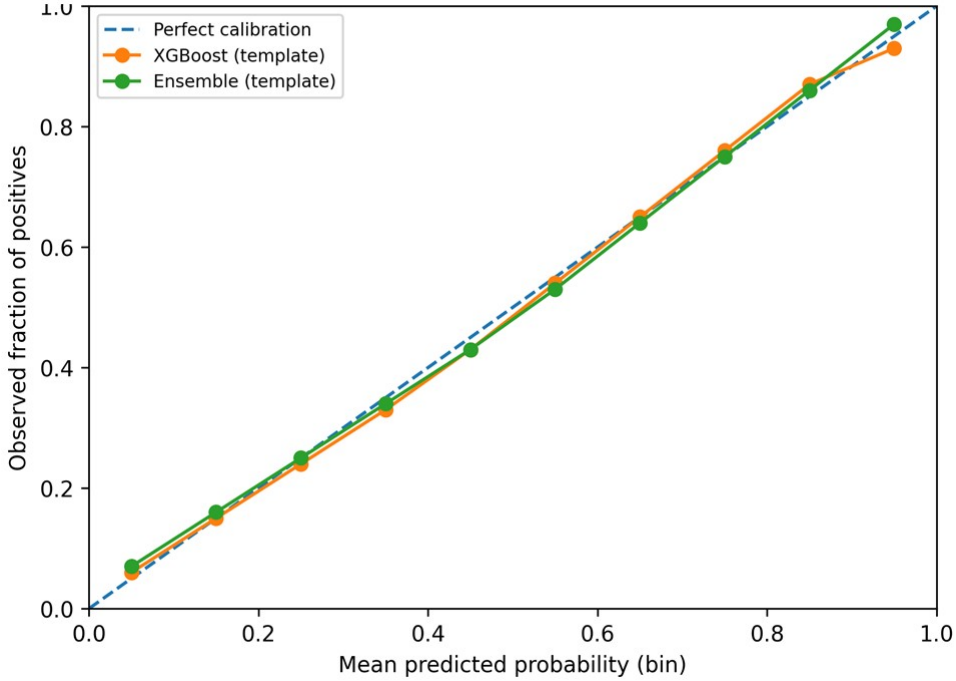


Figure 6. Calibration curves (reliability diagrams) on the held-out time-based test interval for the ensemble and the strongest tabular model.

and risk reduction. This combination of factors points to the system's readiness for further scaling and integration into production management processes, where such forecasts could form the basis for maintenance scheduling, optimisation of spare parts procurement, and improved overall reliability of industrial equipment. To interpret which signals contributed most strongly to the predictive decision, feature-importance analysis was performed for the strongest tabular model. Because XGBoost provided the best standalone tabular performance, its contribution structure was examined using feature-attribution analysis on the held-out time-based test interval. Importance was quantified using mean absolute SHAP values (or gain-based importance if SHAP was not available) on the held-out time-based test interval. Higher values indicate a stronger contribution of a feature to the prediction of pre-failure states. Figure 7 presents the ranking of the most influential predictors used by the tabular branch of the proposed pipeline.

Figure 7 indicates that the predictive signal was not driven by a single sensor channel, but by a combination of vibration-sensitive and process-related descriptors. Features related to vibration amplitude, variability, and short-term trend were expectedly dominant, which is consistent with the mechanical nature of early degradation in rotating and load-bearing components. At the same time, temperature and pressure descriptors contributed complementary information, suggesting that the strongest predictions emerged when mechanical irregularities coincided with broader process deviations. This result supports the multi-sensor design of the proposed pipeline and improves the interpretability of the tabular branch. To better understand model limitations beyond aggregate metrics, a structured error analysis of misclassified cases was conducted. False positives and false negatives were grouped into representative operational categories in order to identify typical failure modes and deployment-related sources of uncertainty. Table 9 presents the main categories of misclassified cases and their likely explanations.

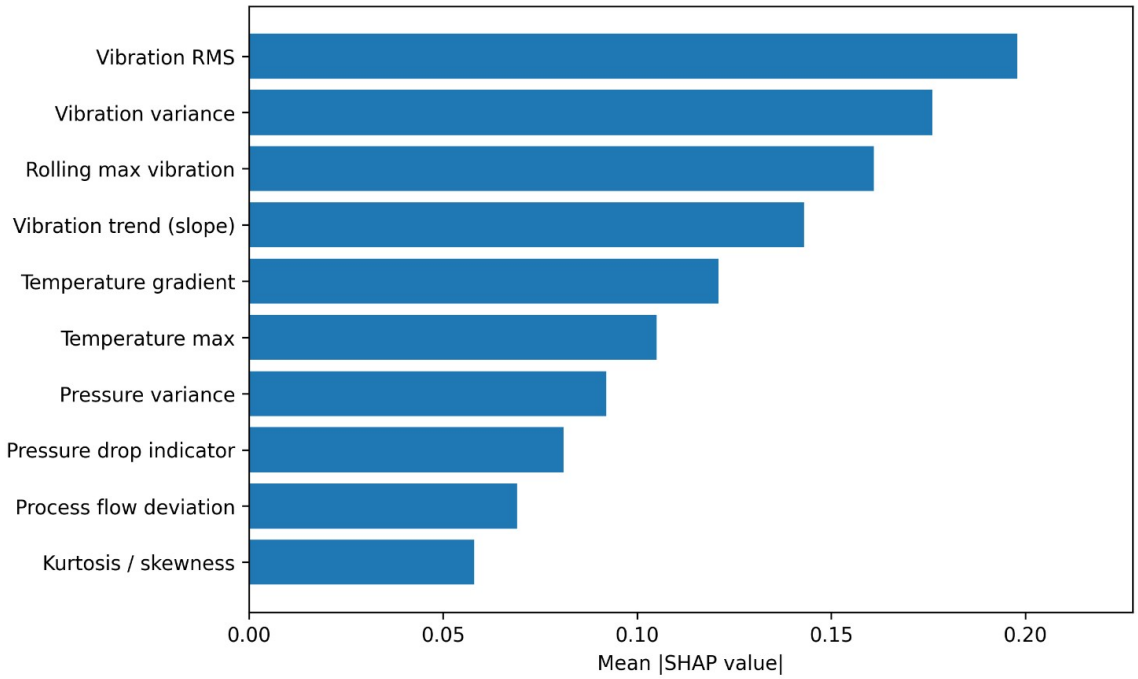


Figure 7. Feature importance ranking for the strongest tabular model (XGBoost).

Table 9. Summary of representative misclassified cases and likely failure modes.

Error category	**Error type**	**Typical signal pattern**	**Likely explanation**	**Operational implication**
Load-change anomaly	False positive	Short-term spike in vibration/variance during rapid regime change	Temporary operating shift rather than progressive degradation	Requires thresholding or context-aware suppression
Maintenance-adjacent anomaly	False positive	Elevated vibration or temperature shortly before/after planned intervention	Signal distortion due to service activity or transient restart effects	Suggests exclusion/masking of maintenance-adjacent windows
Abrupt mechanical failure	False negative	Weak or short precursor signature before failure event	Failure develops too quickly for the selected horizon (H)	May require shorter horizon or faster update interval
Missing-data-related miss	False negative	Gaps or weak sequential continuity in critical window	Missing or degraded sensor information reduces temporal evidence	Highlights need for robust missing-data policy
Borderline degradation case	False positive / uncertain	Moderate anomaly score near threshold (τ)	Early wear indication that does not escalate immediately	Operationally acceptable if preventive inspection is low-cost

Table 9 shows that misclassifications were not random, but were concentrated in a limited number of operationally interpretable scenarios. False positives were often associated with rapid regime changes or maintenance-adjacent signal distortion, whereas false negatives were more likely when degradation signatures were abrupt, weak, or partially obscured by missing data. This pattern suggests that future performance gains may come not only from increasing model complexity, but also from refining the horizon definition, context filtering, and maintenance-aware masking strategy. Therefore, error analysis provides an important complement to aggregate metrics by identifying where the current predictive logic is most vulnerable in deployment.

3.4. Economic efficiency and implementation assessment

In analysing the outcomes of predictive maintenance implementation at SaryarkaAvtoProm LLP, particular attention was paid to assessing financial results and the impact on equipment downtime. Considered were direct costs of planned and unplanned repairs, losses from non-productive downtime, as well as the cost of resources engaged in urgent procurement and emergency work of repair teams. The comparative cost profile for different maintenance strategies is presented in Table 10.

Table 10. Comparative cost analysis of different maintenance strategies.

Maintenance strategy	**Total cost (CU)**	**Unscheduled downtime (hours)**	**Downtime reduction (%)**	**Cost savings vs. Reactive (%)**
Reactive maintenance	1,000,000	340	–	–
Predictive (ensemble model)	720,000	200	$\sim$41	$\sim$28

According to the final data, the transition to predictive maintenance using ensemble algorithms reduced total repair and maintenance costs from 1,000,000 to 720,000 units, corresponding to a saving of approximately 28%. At the same time, the duration of unplanned downtime decreased by about 41% owing to early detection of potential failures and more precise planning of repair operations. Several key aspects served as the main factors behind these savings. First and foremost, advance repair planning played a crucial role. The system predicted the probability of component failure, which made it possible to schedule maintenance in advance and adjust the production timetable to minimise losses. The economic outcomes reported in this section were linked to the operational alert rule derived from the ensemble formulation, namely the generation of a maintenance signal when $\hat{y}(t) = \mathbb{I}[p_{\text{ens}}(t) \geq \tau]$. In other words, the cost reductions and downtime effects were interpreted as downstream consequences of the mathematically defined prediction-and-threshold mechanism rather than of model accuracy alone.

Furthermore, the significant reduction in emergency situations became another source of cost savings. Alerts regarding progressive wear reduced the risk of sudden failures, which in turn lowered the expenses associated with urgent breakdown elimination. Such unplanned repairs are typically far more expensive and result in prolonged downtime, adversely affecting production processes. To position the present study within the current predictive-maintenance literature, a structured comparison of representative related works was compiled. Table 11 summarises the main methodological directions, data types, reported strengths, and limitations of selected studies that are most relevant to automotive and industrial predictive maintenance.

Table 11 indicates that prior studies address important but often separate components of predictive maintenance, such as transfer learning, sequence modelling, economic justification, or maintenance planning. However, few studies combine synchronised tabular and sequential inputs, leakage-free temporal validation, explicit probability-level ensemble fusion, and deployment-oriented evaluation in a single automotive production setting. This comparison clarifies the methodological niche of the present study and explains why additional emphasis was placed on timestamp alignment, class imbalance handling, threshold calibration, and real-time monitoring feasibility. The optimisation of spare part inventories in warehouses was also achieved. The plant was able to obtain advance information on anticipated replacements and replenish stocks of required components in a timely manner. This avoided the need for urgent procurement at inflated prices and prevented supply disruptions, thereby ensuring the uninterrupted operation of equipment and the production process as a whole.

Table 11. Summary of representative related works and their methodological limitations.

Study	**Industry / data context**	**Core method**	**Main strength**	**Limitation relative to the present study**
Shah [1]	Automotive / sensor-based predictive maintenance	ML on sensor data	Early-failure prediction and cost reduction	Limited discussion of heterogeneous multi-rate data alignment and deployment-oriented validation
Wan et al. [4]	Cross-domain fault diagnosis / heterogeneous sources	Transfer learning	Adaptation to heterogeneous operational environments	Does not focus on automotive line-specific synchronisation and threshold-based deployment
Borré et al. [5]	Machine fault detection / time-series signals	CNN–LSTM with attention	Strong temporal feature extraction	Primarily sequence-focused; does not integrate aligned tabular predictors in a single deployment pipeline
Jain et al. [6]	Vehicle health / review study	Systematic review of ML methods	Broad synthesis of predictive-maintenance methods	Review-level evidence; no unified experimental pipeline for automotive production lines
Chen and Wang [7]	Multi-component maintenance planning	Deep reinforcement learning	Optimisation of maintenance decisions	Focuses on planning rather than synchronised fault prediction from sensor streams
Estella [8]	Big data analytics for predictive maintenance	Data-driven analytics	Highlights economic role of data infrastructure	Limited architectural and validation detail for reproducible industrial ML pipelines
Taş [9]	Industry 4.0 / automotive case framing	Predictive maintenance case analysis	Strong practical relevance for automotive competitiveness	Limited methodological detail on ensemble fusion, time-aware validation, and uncertainty handling

Source: compiled by the authors based on [1,4-9].

At the same time, the pilot stage should be interpreted as an initial deployment-oriented validation rather than as a full long-term assessment of model durability. Although the underlying dataset covered a twelve-month observation period and therefore included seasonal and operational variability in the historical records, the online pilot itself was limited in duration and was not sufficient to quantify long-term performance degradation, drift effects, or seasonal changes in alert reliability under continuous real-time use. These outcomes became attainable owing to the system's ability to detect early signs of mechanical degradation and its integration with local maintenance planning tools. Critical alerts were taken into account when scheduling crew tasks and generating spare part requests, enabling preventive maintenance to be carried out during periods of lower production load or allowing the necessary components to be reserved in advance. The shift from emergency fault rectification to proactive diagnostics proved to be the key to improving overall economic efficiency.

In addition to reducing direct expenditure and downtime, the implementation of the predictive strategy contributed to safer working conditions and lowered the risk of damage to large-scale industrial equipment. In cases where abnormal vibration or temperature fluctuations occurred, staff could promptly inspect assemblies and replace worn components before an emergency situation arose. Thus, both technological safety and the overall culture of maintenance planning were enhanced.

The rapid reduction in costs was also reflected in the payback period of the new system. According to the company's financial reports, investment in the installation of additional sensors and the creation of an infrastructure for big data analytics was fully recovered within ten months. This figure took into account not only the procurement costs of IoT equipment but also expenses related to the integration of software solutions, staff training, and the initial deployment of the system on production lines.

Accordingly, a combination of factors – early failure warnings, reduced workload for maintenance personnel, and optimised procurement – resulted in a marked increase in production profitability. Despite the initial expenditure associated with the purchase and installation of sensor arrays and the modernisation of digital infrastructure, costs were recouped relatively quickly due to fewer emergency incidents and improved equipment utilisation efficiency. The analysis confirmed that the transition to predictive maintenance is economically viable and delivers a long-term effect in the form of reduced direct costs, decreased unplanned downtime, and enhanced competitiveness.

4. Discussion

The reported results showed that predictive maintenance in the automotive industry should be interpreted not only through classification accuracy, but through the combined effects of data synchronisation, class-aware thresholding, model fusion, and deployment feasibility. The ensemble of Random Forest, XGBoost, and LSTM achieved the strongest balance between recall and precision, while the pilot monitoring stage demonstrated that the predictive signal was operationally meaningful rather than purely statistical. The reduction in maintenance costs and unplanned downtime therefore depended on methodological coherence across data preparation, time-aware validation, aligned fusion, and threshold-based alerting.

Matzka [10] emphasised that explainability is essential for industrial trust in AI-based maintenance systems. This position was consistent with the reported results, because the predictive pipeline relied not only on final scores, but also on interpretable tabular features, calibration-oriented evaluation, and explicit threshold selection, which made the output more actionable for engineers responsible for maintenance decisions.

This perspective is reinforced by the recent survey of Cummins et al. [11], which showed that explainability in predictive maintenance is not only a transparency issue, but also a practical requirement for trust, adoption, and maintenance decision support in industrial environments.

Yitmen et al. [12] showed that multisensory digital monitoring is most effective when heterogeneous signals are structured into a coherent analytical workflow rather than processed as isolated channels. This logic was reflected in the present results, where the value of vibration, temperature, pressure, and process-flow signals emerged from their alignment on a shared time grid and their transformation into synchronized tabular and sequential representations. Karandaev et al. [13] argued that predictive quality in industrial diagnostics depends heavily on how the signal-processing pipeline is organised before model fitting. The current findings supported this claim, because the final performance of the ensemble was linked not only to model choice, but also to the use of clipping instead of deletion for outliers, gap-length restrictions for interpolation, and leakage-free scaling based solely on the training interval. Paul [14] highlighted that industrial AI systems gain practical value when they remain adaptable under changing operating conditions. This observation was relevant to the present study, since the methodological design explicitly accounted for multi-shift variability, changing equipment loads, and threshold tuning on chronological validation intervals rather than static operating assumptions.

Polverino et al. [15] showed that the main economic value of predictive maintenance comes from preventing high-cost breakdowns rather than marginally improving routine servicing efficiency. The reported reduction in unplanned downtime and total maintenance cost followed this same pattern, indicating that early pre-failure detection created the greatest savings when it prevented escalation into emergency repair and prolonged line stoppage. Erdoğan et al. [16] demonstrated that vibration- and acoustics-based learning systems can detect subtle degradation patterns before they become visually obvious. This finding aligned with the current results, because the sequence-based branch of the model captured temporal irregularities in high-frequency sensor streams that were less visible in aggregated indicators alone, thereby strengthening early warning performance.

Uaisova et al. [17] showed that deterioration prediction frameworks built on continuous monitoring can be transferred across infrastructure settings when the operational logic of the monitored object is preserved. This cross-domain conclusion was relevant to the current study, because the automotive lines investigated here also required continuous condition tracking under variable loads, and the results suggested that the proposed framework could be adapted to other industrial systems with comparable sensorised assets. Lindskog-Münzing and Prehofer [18] examined federated learning as a privacy-preserving strategy for improving industrial models without sharing proprietary plant-level data. This idea was particularly pertinent here, given that the study was based on confidential operational records and that any future extension to multiple plants would likely depend on distributed learning architectures capable of preserving data privacy while capturing a wider spectrum of degradation patterns.

In a different but closely related dimension, Gültekin et al. [19] emphasized that real-time industrial diagnostics must reconcile fast response with computational practicality, especially in settings where alerts are expected under strict operational constraints. That consideration was embedded in the present system design itself: threshold-based alerting, limited inference time, and fixed aggregation windows were treated as core methodological parameters from the outset.

The findings of Nainggolan et al. [20] pointed to the advantage of hybrid analytical frameworks over solutions based solely on either traditional methods or data-driven modeling. A similar tendency emerged in the current study, where the strongest predictive performance was achieved not by a single model class, but through the integration of engineered tabular features, sequential representations, and a soft-voting fusion scheme. Attention to system architecture was also evident in the work of Bajic et al. [21], who underlined the role of edge-oriented processing in lowering latency and strengthening the responsiveness of industrial monitoring. Within the present research, this observation was especially meaningful because deployment was already conceptualized through the separation of aggregation delay and inference delay, indicating that future gains may depend more on relocating parts of the pipeline closer to the data source than on simply replacing classifiers.

A further methodological perspective was provided by Skowron [22], who demonstrated that transfer learning can reduce retraining demands when equipment properties or operating contexts change. Although the present analysis focused on a single historical setup, this point remained highly relevant for the broader applicability of the proposed system, since its long-term scalability will depend on the extent to which previously learned model knowledge can be adapted to new production lines, sensor configurations, or operating regimes.

Chapelin et al. [23] demonstrated that stable predictive performance in heterogeneous production environments depends not only on model transferability, but also on the presence of explicit mechanisms for drift detection and diagnosis. This consideration is especially important for the present study, since any future deployment across additional production lines and longer operational horizons will require the system to capture shifts in sensor distributions, operating regimes, and degradation dynamics over time. A somewhat different emphasis was offered by Radionov et al. [24], who argued that online operational data provide a more accurate basis for estimating residual equipment condition than maintenance logic tied to rigid schedules. This view aligned well with the current findings, because the proposed predictive framework was built around condition-dependent temporal behaviour rather than fixed service intervals, making earlier and more selective intervention possible.

The contribution of deep sequential modelling was clarified by Alkhawaldeh et al. [25], who noted that such architectures are particularly effective when failure signatures emerge through subtle temporal dependencies rather than isolated static variables. The present results followed the same pattern: the LSTM component captured information that was not fully represented by the tree-based models and, once combined within the ensemble, helped achieve a more balanced trade-off between recall and precision. Schwarz et al. [26], in turn, drew attention to the practical value of data augmentation in predictive-maintenance systems, especially under conditions where labelled failure events remain scarce. For the present research, this argument is methodologically significant because further extension of the pipeline to new equipment types or rarely failing components will likely require augmentation strategies to sustain model robustness without relying on the slow accumulation of real-world breakdown cases. The reported results should also be interpreted in relation to practical industrial alternatives, including commercial predictive-maintenance platforms that typically combine anomaly scoring, dashboard-based monitoring, and rule-driven alerting within integrated software ecosystems. Although the present study did not perform a vendor-specific benchmark, the obtained results suggest that the proposed pipeline is competitive in

terms of methodological transparency and adaptability, because its tabular–sequential fusion, threshold-based alert logic, and explicit preprocessing design are clearly specified rather than hidden inside proprietary implementations. At the same time, this advantage in flexibility is associated with a higher modelling and validation burden, meaning that the gain in predictive performance must be evaluated together with the additional complexity required for deployment, calibration, and long-term support.

A further trade-off concerns computational complexity and scalability. The proposed ensemble is more demanding than a single tabular classifier because it requires parallel handling of aggregated features, sequential windows, and probability-level fusion, which increases engineering effort even when inference latency remains operationally acceptable. In practical deployment, scalability will depend not only on model accuracy, but also on update strategy, including periodic revalidation, threshold re-tuning, drift monitoring, and retraining when sensor distributions or operating regimes change. Therefore, the long-term value of the system should be interpreted as a balance between stronger predictive performance and the organisational capacity to maintain the pipeline under evolving production conditions.

The practical deployment of predictive-maintenance systems in industrial environments also raises important data privacy and security considerations. Although the present study relied on internal operational data rather than personal information, sensor streams, maintenance logs, and equipment-failure records may still contain commercially sensitive patterns related to production regimes, process stability, and asset utilisation. For this reason, secure storage, controlled access, and restricted sharing of raw industrial data should be treated as an essential part of deployment architecture rather than as an external administrative issue. In this context, the confidentiality constraints noted in the present study are not only a limitation for reproducibility, but also a realistic reflection of the data-governance requirements that accompany industrial AI adoption.

Ethical considerations are also relevant when predictive-maintenance systems are introduced into real production practice. The purpose of such systems should not be interpreted as replacing maintenance personnel, but as improving the quality and timeliness of technical decision-making by providing earlier and more structured risk signals. In practice, this means that human engineers remain responsible for diagnosis confirmation, intervention planning, and safety-critical decisions, while the predictive model functions as a decision-support instrument rather than an autonomous authority. From this perspective, the workforce impact of the proposed pipeline is best understood as a shift toward more informed and preventive maintenance practice rather than as simple labour substitution.

Taken together, these studies confirmed that the reported results should be interpreted as the outcome of a coordinated methodology rather than the isolated success of a single algorithm. The strongest contribution of the present work lies in combining synchronized heterogeneous inputs, time-aware validation, probability-level fusion, and deployment-oriented evaluation within an automotive production setting. At the same time, the discussion also indicates that further gains will depend on more explicit treatment of uncertainty, broader multi-plant validation, stronger calibration control, and tighter integration between predictive signals and maintenance-planning systems.

The study also had several limitations that should be taken into account when interpreting the reported results. First, the empirical basis was derived from one automotive production site, which limits external generalisability to plants with different equipment architectures, maintenance policies, and sensor infrastructures. Second, part of the predictive pipeline was evaluated under a pilot deployment horizon that was sufficient to demonstrate practical feasibility, but not long enough to quantify long-term degradation of predictive performance, seasonal shifts in alert behaviour, or drift-related instability during continuous operation. Third, some operationally important scenarios, such as abrupt failures with weak precursors or maintenance-adjacent transients, remained difficult to classify reliably, as shown by the error analysis. Finally, the use of confidential industrial data restricted the possibility of public data and code release, which may limit direct replication despite the detailed methodological description provided in this manuscript.

These limitations define several directions for future research. A natural next step is multi-site and longer-term validation across additional automotive lines and plants in order to test the robustness of the proposed pipeline under broader operating variability. Further work should also include explicit drift detection, monitoring of performance degradation over time, seasonal recalibration analysis, and adaptive threshold selection to maintain prediction quality during extended deployment. From the modelling perspective, future studies may explore

richer multimodal inputs, including acoustic sensing and computer vision, as well as data augmentation and transfer-learning strategies for low-failure regimes. In addition, stronger integration between predictive outputs and maintenance-planning systems would make it possible to move from fault anticipation alone towards adaptive scheduling and decision support at the plant level.

5. Conclusions

The study showed that the effectiveness of predictive maintenance in automotive production depends not on the isolated use of one model family, but on the coordinated interaction of aligned heterogeneous data, time-aware validation, and probability-level model fusion. The strongest result was achieved by the ensemble combining Random Forest, XGBoost, and LSTM, which outperformed the best standalone model and provided the most balanced trade-off between recall and precision under the imbalanced pre-failure detection setting. This finding suggests that predictive value emerged from complementarity between aggregated tabular descriptors and sequential temporal patterns rather than from model complexity alone.

An important methodological insight of the study is that the reliability of predictive-maintenance results depends strongly on how the full analytical pipeline is constructed. The reported gains were supported by leakage-free chronological splitting, explicit synchronization of tabular and sequential inputs at a shared endpoint timestamp, horizon-based label assignment, and threshold-based ensemble alerting. In this sense, the contribution of the work lies not only in the achieved metrics, but also in demonstrating a reproducible deployment-oriented framework that links industrial sensor data to operational decisions in a consistent way.

The empirical findings confirmed that the proposed system was practically meaningful under real production conditions. On the held-out test interval, the ensemble reached Accuracy = 0.95, F1 = 0.91, and ROC AUC = 0.96, while pilot deployment showed that its outputs could be used for online risk signalling and maintenance support. The economic analysis further indicated that predictive maintenance based on the developed pipeline reduced total maintenance costs by approximately 28% and unplanned downtime by about 41%, which means that the predictive signal had measurable operational value beyond statistical model performance.

Taken together, these results indicate that predictive maintenance in the automotive industry should be interpreted as an integrated decision-support problem rather than only as a classification task. The practical significance of the study lies in showing that early fault anticipation becomes most useful when probability estimates, threshold calibration, and monitoring logic are aligned with maintenance priorities and production constraints. At the same time, the work also points to the next stage of development: broader multi-site validation, stronger drift handling, richer multimodal data integration, and tighter coupling between predictive outputs and adaptive maintenance planning.

REFERENCES

1. C. V. Shah, *Machine learning algorithms for predictive maintenance in autonomous vehicles*, International Journal of Engineering and Computer Science, vol. 13, no. 1, pp. 26015–26032, 2024.
2. Y. Mahale, S. Kolhar, and A. S. More, *Enhancing predictive maintenance in automotive industry: Addressing class imbalance using advanced machine learning techniques*, Discover Applied Sciences, vol. 7, p. 340, 2025.
3. A. de Giorgio, G. Cola, and L. Wang, *Systematic review of class imbalance problems in manufacturing*, Journal of Manufacturing Systems, vol. 71, pp. 620–644, 2023.
4. L. Wan, J. Ning, Y. Li, and C. Li, *Efficient cross-domain fault diagnosis via distributed multi-source domain deep transfer learning*, Measurement Science and Technology, vol. 36, no. 1, p. 016165, 2024.
5. A. Borré, L. O. Seman, E. Camponogara, S. F. Stefenon, V. C. Mariani, and L. Coelho, *Machine fault detection using a hybrid CNN-LSTM attention-based model*, Sensors, vol. 23, no. 9, p. 4512, 2023.
6. M. Jain, D. Vasdev, K. Pal, and V. Sharma, *Systematic literature review on predictive maintenance of vehicles and diagnosis of vehicle's health using machine learning techniques*, Computational Intelligence, vol. 38, no. 6, pp. 1990–2008, 2022.
7. J. Chen and Y. Wang, *A deep reinforcement learning approach for maintenance planning of multi-component systems with complex structure*, Neural Computing and Applications, vol. 35, no. 21, pp. 15549–15562, 2023.
8. L. Estella, *Leveraging big data analytics for predictive maintenance*, International Journal of Progressive Research in Engineering Management and Science, vol. 4, no. 10, pp. 759–764, 2024.

9. Ü. Taş, *Advancing predictive maintenance: A comprehensive case study through industry 4.0*, International Journal of Automotive Engineering and Technologies, vol. 13, no. 3, pp. 133–142, 2024.
10. S. Matzka, *Explainable artificial intelligence for predictive maintenance applications*, in Proc. Third International Conference on Artificial Intelligence for Industries, Irvine, CA, USA, 2020, pp. 69–74.
11. L. Cummins, A. Sommers, S. B. Ramezani, S. Mittal, J. E. Jabour, M. Seale, and S. Rahimi, *Explainable predictive maintenance: A survey of current methods, challenges and opportunities*, IEEE Access, vol. 12, pp. 57574–57602, 2024.
12. I. Yitmen, H. Sadri, and A. Taheri, *AI-driven digital twins for predictive operation and maintenance in building facilities*, in I. Yitmen, Ed., Cognitive Digital Twins for Smart Lifecycle Management of Built Environment and Infrastructure. Boca Raton: CRC Press, 2023, pp. 65–78.
13. A. S. Karandaev, I. M. Yachikov, A. A. Radionov, I. V. Liubimov, N. N. Druzhinin, and E. A. Khramshina, *Fuzzy algorithms for diagnosis of furnace transformer insulation condition*, Energies, vol. 15, no. 10, p. 3519, 2022.
14. J. Paul, *Predictive analytics in the automobile industry: Driving efficiency and customer satisfaction*, 2022.
15. L. Polverino, R. Abbate, P. Manco, D. Perfetto, F. Caputo, R. Macchiaroli, and M. Caterino, *Machine learning for prognostics and health management of industrial mechanical systems and equipment: A systematic literature review*, International Journal of Engineering Business Management, vol. 15, 2023.
16. F. A. Erdoğan, A. Kucukmanisa, and Z. H. Kilimci, *Detection of fault from acoustic signals in automobile engines using deep learning techniques*, Kocaeli Journal of Science and Engineering, vol. 6, no. 2, pp. 148–154, 2023.
17. M. Uaisova, B. Zharlykassov, D. Aldasheva, A. Artykbayeva, and P. Radchenko, *The use of ANN and machine learning algorithms to predict road surface deterioration*, GEOMATE Journal, vol. 27, no. 121, pp. 136–143, 2024.
18. E. W. Lindskog-Münzing and C. Prehofer, *Federated learning for automotive applications*, Journal of Highway and Transportation Research and Development, vol. 19, pp. 1–6, 2023.
19. Ö. Gültekin, E. Cinar, K. Ozkan, and A. Yazici, *Real-time fault detection and condition monitoring for industrial autonomous transfer vehicles utilizing edge artificial intelligence*, Sensors, vol. 22, no. 9, p. 3208, 2022.
20. N. Nainggolan, E. M. Nima, N. Nainggolan, and B. M. Alwadi, *Advancements in optimization for automotive manufacturing: Hybrid approaches and machine learning*, International Journal of Industrial Engineering and Management, vol. 15, no. 3, pp. 254–263, 2024.
21. B. Bajic, N. Suzic, N. Simeunovic, S. Morača, and A. Rikalović, *Real-time data analytics edge computing application for Industry 4.0: The Mahalanobis-Taguchi approach*, International Journal of Industrial Engineering and Management, vol. 11, no. 3, pp. 146–156, 2020.
22. M. Skowron, *Transfer learning-based fault detection system of permanent magnet synchronous motors*, IEEE Access, vol. 12, pp. 135372–135389, 2024.
23. J. Chapelin, A. Voisin, B. Rose, B. Iung, L. Steck, L. Chaves, M. Lauer, and O. Jotz, *Data-driven drift detection and diagnosis framework for predictive maintenance of heterogeneous production processes: Application to a multiple tapping process*, Engineering Applications of Artificial Intelligence, vol. 139, p. 109552, 2025.
24. A. A. Radionov, I. V. Liubimov, I. M. Yachikov, I. R. Abdulveleev, E. A. Khramshina, and A. S. Karandaev, *Method for forecasting the remaining useful life of a furnace transformer based on online monitoring data*, Energies, vol. 16, no. 12, p. 4630, 2023.
25. L. Alkhawaldeh, M. Zaheeruddin, and J. Pu, *A survey of applying deep learning to fault diagnosis and fault tolerance for automotive vehicles*, 2023.
26. A. Schwarz, J. R. Rahal, B. Sahelices, V. Barroso-García, R. Weis, and S. Duque Antón, *Data augmentation in predictive maintenance applicable to hydrogen combustion engines: A review*, Artificial Intelligence Review, vol. 58, p. 32, 2025.