E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia Science & Technology Asia
Vol.31 No.3 July - September 2026 Page: [278-308]
Original research article

Machine Learning-Based Time-to-Failure Prediction for a Beverage Packaging Labeling Machine

Sarocha Kaewpradab[1], Wanatchapong Kongkaew[2,*]

[1]*Industrial and Systems Engineering Program, Department of Industrial and Manufacturing Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla 90110, Thailand*
[2]*Department of Industrial and Manufacturing Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla 90110, Thailand*

Received 22 June 2026; Received in revised form 25 September 2026
Accepted 27 September 2026; Available online 30 September 2026

ABSTRACT

Predictive maintenance applies machine learning (ML) to forecast industrial time-to-failure (TTF), but reliable evaluation is challenged by changing operating conditions, repeated failure events, and information leakage. This study develops a leakage-aware TTF prediction framework using four years of operational and maintenance data (2021–2024) from an industrial beverage-packaging labeling machine, including 1,560 operational records and 479 unplanned failures. Six ML algorithms—Random Forest (RF), Gradient Boosting, XGBoost, Support Vector Regression, Multi-Layer Perceptron, and CatBoost—were compared with five historical, temporal, persistence, and linear baselines. CatBoost-Cox, using a Cox survival objective, was also evaluated for relative failure-risk discrimination. Nested walk-forward validation with event-grouped partitioning and fold-specific preprocessing, feature selection, and hyperparameter tuning was employed. All ML models outperformed baselines in RMSE, MAE, and MedAE. RF achieved the lowest mean RMSE (1.8813 days), MAE (1.0121 days), and MedAE (0.2552 days), while CatBoost achieved the highest mean R^2 (0.6280). However, no significant RMSE differences remained after Holm correction. CatBoost-Cox achieved a mean concordance index of 0.9155 (SD = 0.0336). Residual, temporal, computational, and sensitivity analyses revealed model-specific trade-offs, emphasizing leakage-aware temporal validation and application-specific model selection.

Keywords: Beverage packaging; Ensemble learning; Machine learning; Predictive maintenance; Time to failure

***Corresponding author:** wanatchapong.k@psu.ac.th doi.org/10.70730/sta.v31i3.268415

1. Introduction

Manufacturing industries are increasingly challenged by the need to find a suitable balance between high equipment reliability, high productivity, and low downtimes for maintenance purposes. In high-speed configuration systems, such as beverage packaging systems, even a small machinery default can bring considerable loss and inefficiency in production. This is the purpose of predictive maintenance (PdM), which helps anticipate the object of failure by using data-driven techniques, which further acts as an agent in the framework of Industry 4.0 [1]. Unlike reactive or time-based maintenance, ML-enabled PdM transforms real-time condition information into actionable maintenance choices. Therefore, it will rely on the reliability of prediction models, but also on the trade-offs of costs concerning the missed failures and associated false alarms, and eventually on the initial investment in digitalization. Therefore, the optimal cost-oriented PdM frameworks combine the implementation decisions and maintenance cost structures, such as the machine learning classification outputs with ROC-based threshold optimization to achieve an overall minimum maintenance cost [2].

Recent advances in ML-based predictive maintenance have greatly enhanced a substantial number of industrial problem areas, such as fault detection, early failure prediction, and decision support aimed at maintenance decisions across several industrial applications [3, 4]. Ensemble, kernel-based, and deep learning models, such as Random Forest, support vector-based methods (SVM/SVR), and deep neural networks, have shown great ability in capturing nonlinear relationships from sensor-driven industrial data and supporting real-time monitoring and alerting systems [5, 6]. Furthermore, IoT infrastructure integration supports data accessibility, ensuring the continuous monitoring of systems and scalability of data-driven analytics in industrial production systems [7]. All these advancements eventually concur with moving from a static maintenance planning scenario to an adaptive, data-driven decision support framework, which is designed to better represent the operational system conditions that evolve continuously [8].

Despite these advancements, serious methodological limitations persist. With many studies, the integration of ML-based prediction models into broader maintenance scheduling or cost-optimization frameworks exists without systematically isolating and evaluating their individual predictive performance. As a result, the comparative robustness, stability, and generalization of different ML models across industrial scenarios are insufficiently explored [1, 7]. Further differences arise because of dataset characteristics, metrics of evaluation (such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2), and conflicting or non-comparable findings among the studies with different experimental protocols [2, 6]. The absence of standardized benchmarks restrains the practitioners' selection of informed data evaluation at the start of predictive maintenance implementation, especially for complex continuous production systems like the beverage packaging line [4, 5].

This research study addresses those gaps by systematically presenting a comparison of classical and ensemble ML models, including Gradient Boosting, XGBoost, Random Forests, SVR, and Multi-Layer Perceptron for predicting the time to fail-

ure (TTF) in an industrial beverage packaging system. TTF is seen as a continuous, time-based target variable in the proactive maintenance task of objective-time estimation of the remaining useful life for equipment in any continuous production environment. Through the presentation of a common experimental framework, the use of multiple complimentary measures to evaluate provided the algorithmic benchmarking map, which describes all these advantages, disadvantages, and behavioral characteristics of various machine learning paradigms. It sets main direction and significance of the research in the context of the evaluation of TTF predictive models on real-world industrial data circumstances, which may lay down the foundations for theoretical as well as practical implications in the decision-making of the best-fit, reliable, and robust models. The study ultimately paves the way for the construction of models toward data-driven reliability engineering and predictive maintenance in manufacturing systems.

The remainder of this paper is organized as follows. Section 2 reviews machine-learning-based predictive maintenance, TTF prediction, and related industrial applications. Section 3 describes the industrial dataset, feature engineering, predictive models, nested walk-forward validation, and evaluation procedures. Section 4 presents predictive performance, temporal and residual analyses, computational performance, statistical comparisons, and sensitivity analysis. Section 5 summarizes the main findings, limitations, and directions for future research.

2. Literature Review

2.1 Predictive maintenance in the context of industry 4.0

Predictive maintenance (PdM) is a core aspect of the Industry 4.0 paradigm, the models for early failure prediction and, with respect to the degradation that eventually existing equipment, and early diagnosis of potential failures are feasible through the incorporation of IoT, machine learning, and data analytics. The adoption of traditional reactive and calendar-based preventive maintenance strategies means managing increasingly complex and data-rich industrial environments is often too challenging. The PdM frameworks, therefore, employ cyber-physical systems and handle intelligent data fusion for real-time monitoring of asset condition and maintenance scheduling [9].

PdM supports manufacturers in improving availability and reducing unplanned downtimes-increasing efficient reliability performance. An Industry 4.0 system is where new analytical approaches are integrated into the very workings of manufacturing, and a characteristic of such systems is the expected transformation in the way maintenance as an organization and service is matured as decision-based data support that enhances reliability [10]. For an effective application, cleaner data management is needed for perfect fitting, and this turns out to be of importance essentially in businesses such as fast-cycle production and asset heterogeneity. The PdM shows the path to good outcomes because data exchange is necessary in broad terms between disparate legacy systems and the existing information systems, because the problems with capacities and real-time processing will spoil the benefits of PdM themselves [11].

2.2 Machine learning approaches for reliability and Mean Time Between Failures (MTBF) prediction

Machine Learning is transforming reliability engineering, as it finds hidden

patterns in large industrial datasets with many variables which would not be captured by traditional statistical techniques. In support of data-driven reliability evaluation and maintenance decision-making across various industrial sectors [12], some popular and widely used failure classification or regression tasks for machine learning include algorithms such as Random Forest (RF), Gradient Boosting, and SVM.

Aminzadeh et al. [13] implemented predictive maintenance for industrial compressors using several sensors supported by IoT technologies and machine learning. A regression model was proposed for live monitoring and threshold alerts to enhance the responsiveness of maintenance. Their findings showed that condition monitoring supported by data reduces unplanned downtime and enhances equipment reliability by detecting abnormal operational trends early. Similarly, Yadav et al. [14] explain that the XGBoost and LSTM outperform the techniques in the prediction of machine system failures on imbalanced industrial datasets. These models capture nonlinear and temporal degradation. The degradation given cannot be identified by traditional approaches. Another set of works provides some ideas about predicting the remaining useful life (RUL) that are relevant in the concept of reliability metrics.

Real-time RUL estimation models in manufacturing industries have developed to make a comparison of RF and XGBoost under different preprocessing strategies and show that data preparation quality influences performance [15]. In addition, recent predictive maintenance frameworks integrate gradient boosting algorithms like CatBoost to estimate component wear and operational failure limits [16], while extended survival models such as Cox regression capture the complex influence of environmental and operational covariates on equipment lifetime [17]. Taken together, these studies will show that robustness and practical interpretability are essential for deploying data-driven models of reliability into industrial operations and maintenance contexts in the real world.

2.3 Evaluating machine learning models in industrial applications

When reliability prediction through machine learning approaches is considered, a higher predictive accuracy combined with real-time responsiveness, interpretability, and scalability is important. Varalakshmi and Kumar [18] illustrated that combining deep reinforcement learning with ensemble techniques improves real-time fault prediction in Industrial IoT data streams by continuously adapting policy learning to dynamic operating conditions.

The framework of evaluation is currently more oriented towards the industrial key performance indicators, such as false-positive rate, latency, and error-based metrics (e.g., RMSE), along with the production-oriented measurements. García and Salgado [19] conducted a case study in a multi-stage manufacturing system to demonstrate that a decision-oriented predictive maintenance framework with monitoring of the digital behavioral twin (DBT) could reduce unplanned stoppages and strategic maintenance adjustments if failures occurred before the MTTF.

Interpretability is also a significant challenge. Maintenance engineers have to come to trust and understand the algorithmic recommendations before they integrate them within scheduling systems. Hence, the method of explainable AI (XAI), such as SHAP and LIME, is methodically launched to elucidate the variable importance and model rationales. Moreover, model evalua-

tion under data drifts, especially when sensor behavior changes owing to process evolution, becomes essential to maintain long-term accuracy for predictive exercises [20].

Many existing studies aim to evaluate those models under ideal conditions of the laboratory rather than under real-world variability in industrial processes, leaving a gap in how ML models may generalize across different types of equipment and environmental contexts.

2.4 Research gaps and future directions in ML-based TTF prediction for manufacturing systems

Even though predictive maintenance using machine learning has been researched in literature, there are few researches that are specific when it comes to predicting the TTF and MTBF of beverage processing systems. Overall, packaging machines face fundamental difficulties such as high rate of usage, intermittent wearing of components, endurance tests over several introduction practices and lots of changes, and dynamic data acquisition due to these changes. As a result their learning abilities with respect to machine learning (ML) are slow. This means that particular ML frameworks may be unsuitable for the desired modern manufacturing applications.

Future studies should be aimed at building domain-specific TTF datasets involving sensor fusion through the different stations during bottling, labeling, and capping. This marriage of statistical reliability techniques with ML can emerge as a class of hybrids for expressing the uncertainties of predictions for TTF, which is critically important in industrial PdM applications [11].

Explainable ML and digital twin integration further promise boost interpretability and accuracy. For ML to learn from degradation, digital twins can be utilized to simulate degradation under different loads, while transfer or continual learning helps mitigate data scarcities in beverage manufacturing. Trends in data-driven PdM explained certain challenges, as presented in [14].

Finally, creating benchmarking frameworks for TTF forecast will be beneficial in order to compare models objectively and progress toward best practices compliance across the manufacturing sector, similar to those in RUL research.

3. Research Methods

A systematic and data-driven study was carried out to compare various machine learning models applied to the Time-to-Failure (TTF) prediction of an industrial beverage packaging system. In the process, an experimental perspective was employed, and supervised regression was featured as the most successful implementation. Such supervision was adopted to implement uniform and easy to reproduce procedures for each category of six methodologies: Random Forest (RF), Gradient Boosting (GB), Extreme Gradient Boosting (XGBoost), Support Vector Regression (SVR), Multi-Layer Perceptron (MLP), and Categorical Gradient Boosting (CatBoost). These six models were evaluated as direct TTF regression models. In addition, CatBoost was further implemented using a Cox survival objective, hereafter referred to as CatBoost-Cox, to provide a complementary survival-analysis perspective that explicitly accommodates right-censored observations and estimates relative failure risk rather than direct TTF in days. Remaining operational steps are the acquisition of real-world manufacturing data, data cleaning and transformation, model selection with hyperparameter search, and performance assessment based on several metrics. Ac-

cordingly, the six direct regression models and CatBoost-Cox were evaluated using model-specific performance criteria appropriate to their different prediction objectives. This framework implies respect of the need for the development or comparison of predictive maintenance models in the presence of historical and repair information.

3.1 Collection, preprocessing, and feature engineering

The data set that has been employed in this study was gathered from an industrial labeling machine operating within a high-speed beverage bottling and conveying system. It includes equipment-condition records over a four-year period from 2021 to 2024, encompassing both normal operating conditions and failure-related behavior. The raw dataset comprised 1,560 operational records and 479 documented unplanned failure events. Planned maintenance activities and scheduled production stoppages were not classified as failure events. In this study, Time-to-Failure (TTF) was defined as the time interval between each observation and the next recorded unplanned failure.

Each of the 1,560 records represents one timestamped operational observation of the labeling machine. Therefore, the 1,560 records should not be interpreted as 1,560 independent failure events. Multiple operational observations could occur before the same subsequent unplanned failure and could therefore be associated with the same future-failure event while having different Time-to-Failure (TTF) values because their observation timestamps differed. The TTF is computed via Eq. (3.1), as follows:

$$TTF(i) = t_{f(i)} - t_i, \qquad (3.1)$$

where t_i denotes the timestamp of observation i, and $t_{f(i)}$ denotes the timestamp of the next unplanned failure.

TTF, Mean Time Between Failures (MTBF), and Remaining Useful Life (RUL) were treated as distinct reliability concept. In general reliability terminology, TTF refers to the elapsed or remaining time associated with failure occurrence, depending on the specific modeling formulation. In the present study, the term TTF refers specifically to the retained operational target defined above. MTBF represents the average interval between consecutive failure events and was used only as a descriptive reliability statistic. RUL generally describes the estimated remaining operating life of an asset before a predefined end-of-life condition and was not used as the prediction target in this study.

The original TTF target was retained unchanged for the direct regression models (RF, GB, XGBoost, SVR, MLP, conventional CatBoost, and reference baselines). In contrast, CatBoost-Cox used a dedicated survival representation evaluated in parallel under a Cox partial-likelihood objective. Completed operating intervals ending in an unplanned failure were treated as observed failure events, whereas intervals without a subsequent failure prior to the study cutoff were right-censored. Any initial incomplete interval lacking an observed starting point was excluded. Eq. (3.2) defines the resulting signed survival label used by CatBoost-Cox, as follows:

$$y_i^{Cox} = \begin{cases} +d_i, & \delta_i = 1, \\ -d_i, & \delta_i = 0, \end{cases} \qquad (3.2)$$

where y_i^{Cox} denotes the duration of survival episode i and δ_i is the event indicator. Duration is encoded as positive for observed failures ($\delta_i = 1$) and negative for right-censored intervals ($\delta_i = 0$).

This signed survival representation was used exclusively within the CatBoost-Cox branch and did not modify the target

used by the six direct regression models. Because the operational target was reset at a recorded failure event, the duration of each completed survival episode was obtained from the positive elapsed duration immediately preceding the corresponding failure record. The first observed failure interval was excluded from the survival construction when its complete starting point occurred before the beginning of the available observation history. Therefore, the Cox analysis used survival episodes as its analytical unit rather than the individual operational records used by the direct regression models.

The structure of the raw operational dataset structure is illustrated in Table 1, which shows the main data variables mined from the factory logs, such as timestamp, operational time, downtime duration, Cleaning in Place (CIP), and the category-coded symptom descriptors. These variables are the basis for reliability analysis and, hence, machine-learning-based TTF prediction.

Table 1. Raw operational dataset structure.

Variable	Type	Unit	Description
Timestamp	Date/Time	–	Observation timestamp
Operating time	Numeric	hour (h)	Machine operating duration
Downtime	Numeric	minute	Duration of machine stoppage
CIP	Binary	0/1	Cleaning-in-Place status
Downtime/ Failure	Binary	0/1	Occurrence of an unplanned failure
Symptom	Categorical	–	Type of observed anomaly

To account for temporal variability and nonstationary operating conditions, the dataset was summarized by year. Each yearly subset was used for descriptive analysis and year-wise performance assessment. The characters of each year's data set, namely the failure events, number of records, and TTF statistics, are given in Table 2. Monthly variations, however, indicate changes in production load, maintenance intervention, and operational stress, underscoring the need for robust and generalizable models.

The substantial inter-annual variation in MTBF values reflects changes in production load, maintenance interventions, and operational stress. Although the outer loop of the nested cross-validation adopts a walk-forward strategy to preserve temporal ordering and prevent information leakage, the operating conditions of the labeling machine remain inherently non-stationary across years. As a result, predictive performance may vary under different operating regimes.

Table 2. Yearly dataset of labelling machine.

Year	Number of Records	Number of Failure Events	Average MTBF (days)	Std. Dev. of MTBF (days)
2021	369	55	6.0	6.0
2022	401	169	1.8	2.6
2023	392	125	2.2	2.4
2024	398	130	3.9	5.3
Total / Average	1,560	479	3.4	4.1

Standard data cleaning includes robust date parsing, target variable numeric conversion, and deletion of invalid records for data-consistency purposes. Numerical missing values were replaced by median filling, while random variables assigned constant categories when missing. Although extreme values were left alone in the data, they could be representatives of high-dimension pre-failure signals concerning industrial reliability.

All preprocessing steps after data import, namely, imputation, encoding, and scaling, were united under a Scikit-learn pipeline, then cross-fitted with train folds exclusively inside nested walk-forward val-

idation to prevent data leakage. This setup allows guaranteed reproducibility in full play, with rigorous integrity in temporal considerations regarding the evaluation. According to Table 3, while categorical variables were one-hot encoded, with trail categories being ignored in the encoder, quantitative predictors were standardized by Z-score normalization to improve numerical stability finally.

Table 3. Data preprocessing and leakage-prevention strategy.

Step	Method	Purpose
Missing values	Median imputation (numerical), constant category (categorical)	Handle incomplete industrial records
Scaling	#Z-score normalization (StandardScaler)	Improve numerical stability
Categorical encoding	One-hot encoding (ignore unknown categories)	Preserve operational semantics
Outlier handling	Retained as potential pre-failure signals	Preserve failure-related information
Feature election	Mutual information Top-12 ranking, Spearman diagnostic analysis, and iterative VIF filtering	Reduce redundant and collinear predictors within each training fold
Leakage prevention	Pipeline re-fitted per fold (walk-forward nested CV)	Ensure reproducibility and prevent information leakage

To address feature redundancy, feature selection was performed independently within each training fold to prevent information leakage. First, mutual information (MI) was calculated between each candidate predictor and the training target, and the 12 highest-ranked nonconstant features were retained. Spearman correlation was then used as a diagnostic measure to examine monotonic associations among the selected candidates. Finally, iterative Variance Inflation Factor (VIF) analysis was performed, removing the feature with the highest VIF until all retained predictors satisfied VIF < 5. Because this procedure was repeated within each training fold, the selected feature subset was allowed to vary across folds rather than being fixed throughout the analysis. The candidate engineered predictors considered in this fold-specific selection procedure are summarized in Table 4.

These predictors were engineered from structured operational and maintenance records to characterize several aspects of machine behavior, including recent downtime history, time since previous events, symptom occurrence and transitions, and short-term operating trends. Most predictors were derived from historical observations, whereas several symptom-related variables incorporated information available at the current observation. These variables therefore represented the machine condition at the prediction point rather than information from future records.

Because the feature-selection procedure was re-estimated within each training fold, the final predictor subset could adapt to temporal changes in operating conditions. This approach ensured that feature ranking, correlation assessment, and multicollinearity screening were based exclusively on the corresponding training data.

3.2 Machine learning models and method selection

This study evaluates six widely used machine learning models for the TTF results based on the complementary learning paradigms commonly applied in predictive-maintenance studies. The models are RF, GB, XGBoost, SVR, MLP and CatBoost; these models were chosen in a manner to enable the systematic comparison of ensemble-based, kernel-based, and neural network approaches under the identical industrial data conditions. In addition to conventional CatBoost regression,

Table 4. Candidate engineered predictors used for fold-specific feature selection.

Feature	Operational definition
days_since_last_overhaul	Number of ordered observations since the most recent scheduled overhaul indicator
days_since_last_downtime	Number of ordered observations since the most recent downtime event
Downtime_Freq_90	Mean downtime-event indicator over the preceding 90 records
Downtime_count_14d	Number of downtime events over the preceding 14 records
Downtime_sum_30d	Total downtime duration over the preceding 30 records
symptom_count_7d	Number of symptom-positive observations over the preceding 7 records
symptom_score_trend	Difference between the current and immediately preceding symptom-severity scores
Downtime_trend_7d	Linear slope of downtime values over the preceding 7 records
Downtime_volatility	Product of the preceding 30-record downtime sum and the rolling standard deviation of operating time
Downtime_5d_avg	Mean downtime duration over the preceding 5 records
downtime_ratio_30d	Ratio of preceding 30-record downtime to preceding 30-record operating time
Symptom_Freq_30d	Number of symptom-positive observations over the preceding 30 records
Symptom_Freq_30d_ratio	Thirty-record symptom frequency divided by 30
Symptom_Co_occurrence	Indicator that a symptom was recorded at the current observation

CatBoost was separately trained using a Cox survival objective. This additional formulation, termed CatBoost-Cox, was used to investigate relative failure-risk discrimination while accounting for censored survival information. Therefore, conventional CatBoost regression and CatBoost-Cox represent two different learning objectives within the same CatBoost algorithmic family rather than two unrelated algorithms.

3.2.1 Machine learning models

The six machine-learning models were selected because they represent different mechanisms for learning nonlinear relationships between operational characteristics and TTF. All six models, including conventional CatBoost, were evaluated as direct regression models. CatBoost was then additionally formulated using a Cox survival objective to provide a complementary survival analysis perspective. Their conceptual modeling structures are illustrated in Fig. 1.

RF combines multiple decision trees through bootstrap aggregation to reduce prediction variance and improve robustness to noisy data, whereas GB sequentially constructs weak learners to reduce prediction error. XGBoost extends conventional gradient boosting by incorporating regularization and optimized tree construction to improve generalization. SVR represents a kernel-based regression approach capable of modeling nonlinear relationships through implicit feature transformation and regularization. Conventional CatBoost regression employs ordered boosting and symmetric decision trees and was trained using an RMSE loss function to produce direct TTF estimates. MLP represents a feed-forward neural-network approach capable of learning complex nonlinear relationships between operational predictors and the target.

For the additional survival analysis, CatBoost was trained using the Cox objective instead of the conventional RMSE objective. CatBoost-Cox learns a nonlinear risk function within the Cox proportional-hazards framework.

$$h(t|X) = h_0(t) \exp [f(x)], \qquad (3.3)$$

where $h(t|X)$ denotes the hazard conditional on predictor vector x, $h_0(t)$ represents the unspecified baseline hazard, and $f(x)$ represents the nonlinear risk function learned by the CatBoost ensemble. The formulation in Eq. (3.3) therefore combines

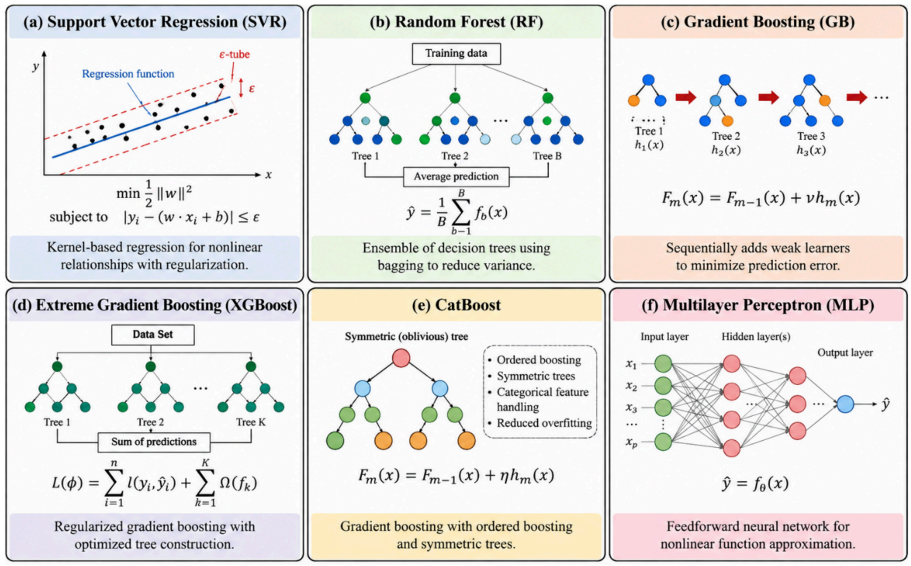


Fig. 1. Structural and mechanical differences in TTF prediction modeling paradigms.

gradient-boosted decision trees with a semi-parametric survival objective capable of incorporating right-censored observations. Unlike conventional CatBoost regression, CatBoost-Cox does not produce a TTF prediction expressed in days. Its output is a relative failure-risk score. Higher scores indicate greater modeled relative failure risk, whereas lower scores indicate lower modeled relative risk. Consequently, CatBoost-Cox was evaluated separately from the six direct regression models using survival-specific metrics.

Together, these models provide complementary modeling structures for evaluating predictive accuracy, robustness, temporal consistency, and computational performance within the same TTF prediction framework. The additional CatBoost-Cox formulation extends this framework by providing a complementary evaluation of failure-risk discrimination and censored survival information without replacing the original CatBoost regression analysis.

3.2.2 Reference baseline models

In addition to the six machine-learning models, five reference baseline approaches were included to determine whether the predictive performance of the proposed models exceeded that of simpler historical, temporal, linear, and reliability-based forecasting methods. The TTF target followed the definition provided in Section 3.1, and $y_i = TTF_i$ denotes the observed target associated with observation i.

Because the TTF value of a historical observation becomes known only after its associated future failure has occurred, only observations with fully observed subsequent failure outcomes were considered eligible for baseline construction. Eq. (3.4) defines the historical information available at prediction time t_i, given by

$$H_i = \{j : t_{f(j)} \leq t_i\}, \qquad (3.4)$$

where j is eligible for training, $t_{f(j)}$ denotes the timestamp of the unplanned failure used to determine the TTF value of observation j. During nested walk-forward

validation, eligible observations were further restricted to the corresponding training partition to prevent future-label information from entering the baseline calculations.

1. **Historical mean TTF**

   The Historical Mean TTF baseline represents the simplest global historical reference. For each outer validation fold, the prediction was calculated as the arithmetic mean of the eligible TTF values available in the corresponding training history via Eq. (3.5), given by

$$\hat{y}_i^{Mean} = \frac{1}{|H_{train}|} \sum_{j \in H_{train}} y_i, \quad (3.5)$$

   where H_{train} denotes the set of eligible labeled observations within the training partition.

   The resulting value was used as a constant prediction for the corresponding outer-test partition and was recalculated when the expanding training window was updated. This baseline provides a basic reference for determining whether the machine-learning models can improve upon the long-term historical average of TTF.

2. **Historical median TTF**

   The Historical Median TTF baseline was used as a robust alternative to the historical mean. The prediction was defined as

$$\hat{y}_i^{Mean} = median\{y_j : j \in H_{train}\}. \quad (3.6)$$

   As with the Historical Mean baseline via Eq. (3.6), the estimated value was held constant throughout the associated outer-test partition and recalculated when a new training window was formed.

   The median is less sensitive to unusually large TTF values or extreme failure intervals and therefore provides a useful reference when the historical TTF distribution is skewed or contains extreme observations.

3. **Temporal moving mean TTF**

   The Temporal Moving Mean baseline was designed to represent a simple adaptive forecast based on recent failure behavior. Let $W_i \subseteq H$ denote the set containing up to the K most recent eligible historical TTF observations available at prediction time t_i The predicted TTF was calculated as

$$\hat{y}_i^{TMean} = \frac{1}{|W_i|} \sum_{j \in W_i} y_j. \quad (3.7)$$

   In contrast to the Historical Mean and Historical Median, this baseline via Eq. (3.7) emphasizes more recent observations and can therefore respond to temporal changes in failure behavior. The window size K should be specified before outer-test evaluation or determined exclusively from the corresponding training data. As new failure outcomes become fully observed, the available temporal window can be updated without using future information.

   This baseline was included to determine whether recent historical TTF behavior alone can capture temporal changes in machine operating conditions without requiring a more complex predictive model.

4. **Previous failure interval**

   The Previous Failure Interval baseline represents a simple reliability-persistence approach based on the duration of the most recently completed interval between two consecutive unplanned failures.

   Let F_{m-1} and F_m denote the timestamps of the two most recent completed failure events known at prediction time t_i. The predicted TTF was defined as

$$\hat{y}_i^{PrevInt} = F_m - F_{m-1}. \qquad (3.8)$$

   The interval was measured using the same time convention as that adopted for the primary TTF target. The baseline in Eq. (3.8) tests whether the most recent completed failure interval can provide useful predictive information without requiring statistical learning or feature-based modeling. Because it predicts the complete duration of the previous failure interval rather than the remaining duration of the current interval, it should be interpreted as a simple persistence benchmark rather than as an exact estimate of remaining useful life.

5. **Linear regression**

   Linear Regression was included as a conventional feature-based predictive baseline. Unlike the four preceding baselines, which rely primarily on historical target information, Linear Regression uses the operational predictors to estimate TTF.

   Given p available predictors, the predicted TTF for observation i was computed via Eq. (3.9), given by

$$\hat{y}_i^{LR} = \hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k X_{ik}, \qquad (3.9)$$

   where X_{ik} denotes the value of predictor k for observation i, $\hat{\beta}_0$ is the estimated intercept, and $\hat{\beta}_k$ represents the estimated regression coefficient for predictor k.

   All data-dependent preprocessing, feature selection, and scaling procedures for Linear Regression were performed exclusively using the corresponding training data, consistent with the leakage-prevention strategy adopted for the machine-learning models. Linear Regression therefore provides a conventional learned benchmark for assessing whether nonlinear models capture predictive relationships beyond those represented by a linear combination of the operational predictors.

   Together, the five baseline approaches provide complementary reference levels with increasing predictive complexity. Historical Mean and Historical Median represent global historical benchmarks, Temporal Moving Mean provides a time-adaptive reference, Previous Failure Interval represents a reliability-based persistence benchmark, and Linear Regression provides a conventional feature-based learned model. Their inclusion enables the six machine-learning models to be evaluated not only against one another but also against substantially simpler forecasting alternatives under the same temporally ordered validation framework.

3.3 Hyperparameter optimization and validation strategy

All machine-learning models were trained and assessed via a common and reproducible experimental protocol, which ensured fair comparison. A nested walk-forward validation strategy was adopted to preserve the chronological structure of the data and to reduce the risk of information leakage. The outer validation consisted of five expanding-window folds and was used for final model evaluation, whereas three inner time-series folds were constructed within each outer-training period for feature selection and hyperparameter optimization. Consequently, each outer-test period represented a later, previously unseen segment of the data and remained excluded from all model-development procedures.

The size of the outer-training set progressively increased as additional historical observations became available. The numbers of training and test records, test horizons, and failure-related observations for each outer fold are summarized in Table 5. The training set increased from 415 records in Fold 1 to 1,289 records in Fold 5, while each fold was evaluated on a distinct subsequent test period. This expanding-window structure allowed model performance to be assessed on temporally unseen observations while progressively incorporating additional historical information into the training set.

Table 5. Outer walk-forward validation structure.

Outer Fold	TrainStart	Train End	Test Start	Test End
1	1/1/2021	11/2/2022	12/2/2022	24/6/2022
2	1/1/2021	24/6/2022	25/6/2022	26/1/2023
3	1/1/2021	26/1/2023	27/1/2023	10/8/2023
4	1/1/2021	10/8/2023	11/8/2023	29/4/2024
5	1/1/2021	29/4/2024	30/4/2024	31/12/2024

To maintain temporal integrity, observations were partitioned according to chronological dates so that records from the same date were not divided between training and test sets. All data-dependent procedures, including missing-value imputation, optional IQR-based treatment of extreme values, mutual-information feature ranking, VIF-based feature screening, scaling, and hyperparameter optimization, were estimated exclusively from the corresponding training data.

After model selection within the inner folds, the complete preprocessing and modeling pipeline was re-fitted using the full outer-training set before predictions were generated for the outer-test set. Records associated with the same subsequent failure event were additionally grouped within the same validation partition to prevent event-related observations from being distributed across training and test sets. No future-event group was shared between the training and test sets in any outer fold (SharedFutureEventGroups = 0), confirming event-level separation throughout the evaluation. The event identifier was used solely for partition control and was not included as a model predictor.

Hyperparameters were optimized using randomized search within the inner time-series validation framework, with up to 30 configurations evaluated per model and outer fold. When fewer than 30 possible configurations were available, all configurations were evaluated. RMSE was used as the optimization criterion, and a fixed random seed of 42 ensured reproducibility. The hyperparameter search spaces are summarized in Table 6.

All experimental modules were conducted on an ASUS TUF Dash F15 model (Intel Core i7-12650H, 16GB RAM RTX 3050 Ti), with Windows 11, Python 3.11, Scikit-learn, and XGBoost and CatBoost. For added consistency in experiments, a

fixed set of random seeds (seed = 42) was used to ascertain a deterministic reproducibility across all repeated trials. Furthermore, a fixed data split and recorded training time enhance computational transparency and reproducibility.

Table 6. Hyperparameter tuning settings.

Model	Key Hyperparameters
RF	n_estimators (50,100,200), max_depth (5,7,10, None), min_samples_split (2–5), min_samples_leaf (1,2,4), max_features (sqrt, log2)
GB	n_estimators (50,100,200), max_depth (3–5), learning_rate (0.03, 0.05, 0.1)
XGBoost	n_estimators (50,100,200), max_depth (3–7), learning_rate (0.01, 0.05, 0.1), subsample (0.8–1.0), reg_lambda (0.1–1.0), colsample_bytree (0.8, 1.0)
SVR	C (0.1–100), ε (0.01–0.5), kernel (RBF), γ (0.01–1.0, scale, auto)
MLP	hidden layers (50 to 100–50–25), activation (ReLU, tanh), learning rate (0.001–0.1), α (0.001–0.1), max_iter (1000–2000)
CatBoost	iterations: {100, 200, 400}; depth: {3, 4, 6}; learning_rate:{0.03, 0.05, 0.1}; l2_leaf_reg: {3, 5, 10}

Although the study was carried out offline using nested walk-forward validation, the training design adopted will apply well in real-world deployments in industrial environments that expect nonstationarity. In practical scenarios, the framework could be enhanced to include sliding-window retraining combined with drift detection techniques such as Adaptive Windowing or ADWIN, enabling automatic model updates in response to distribution shifts or prediction errors, thus preserving the consistency of time-based data.

3.4 Evaluation metrics, comparative analysis, and reproducibility

Performance was evaluated by some complementary metrics that were suitable for different dimensions of measurement and prediction of TTF quality. The Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are prominent tests for the accuracy of predictions, with the RMSE placing more emphasis on large errors, and the MAE giving an overall view of accuracy. Additionally, Median Absolute Error (MedAE) was used to give a more robust measure of typical prediction error and also to pass over very extreme outliers often appearing in failure data from real industrial sectors, while the R^2 statistics were used to assess the descriptive abilities of the regression coefficient and to quantify the goodness of the model fit.

Because CatBoost-Cox produces relative failure-risk scores rather than direct TTF predictions, RMSE, MAE, MedAE, and R^2 were not applied to the Cox formulation. Instead, CatBoost-Cox was evaluated using Harrell's concordance index (C-index), which assesses the agreement between predicted risk ordering and observed survival ordering while incorporating censored information.

Statistical uncertainty for CatBoost-Cox was summarized using the mean and standard deviation of the five outer-fold C-index estimates. A 95% confidence interval was additionally estimated by bootstrap resampling of the fold-level C-index values. An exploratory one-sided Wilcoxon signed-rank test was performed against a reference C-index of 0.50. Because only five outer-fold estimates were available and the expanding training windows overlapped temporally, this hypothesis test was treated as supplementary rather than as definitive inferential evidence.

The comparative analysis sought to assess trade-offs among accuracy, robustness, and generalization capability across models. This would provide the error distribution analysis as well as perform statistical tests to prove if the systemic causes of the studied performance differences resulted from random variations. The model was also subjected to additional treatment beyond sensitivity analysis under differ-

ent data conditions with regard to industrial beverage packaging solutions, which usually work in nonstationary ways. To ensure the transparency and reproducibility of the study, the standard ML evaluation approaches, such as all the steps of pre-processing, hyperparameter tuning, and validation procedures, are explicitly explained. All experiments have been conducted with the identical data partition and subject to predefined random number seeds. Wherever suitable, anonymized training data or model configurations are made available to enable potential replication and benchmarking efforts. This enabled the development of the frame for a simple contrast of the TTF forecast models, ensuring realism and practicality of the results made visible.

4. Computational Results

This section reports and interprets the computational results obtained from the comparative evaluation of six machine learning models: RF, GB, XGBoost, SVR, MLP and CatBoost, for predicting TTF in an industrial beverage packaging system. These six models were evaluated as direct regression models using RMSE, MAE, MedAE, and R^2. In addition, CatBoost was separately implemented using a Cox survival objective (CatBoost-Cox) to provide a complementary evaluation of relative failure-risk discrimination while accounting for right-censored observations. CatBoost-Cox was therefore evaluated separately using the concordance index (C-index) rather than direct TTF prediction-error metrics. This checks predictive efficiency, how the model deal with the variability in the data, error behavior, survival-risk discrimination, and the computational feasibility. Because CatBoost regression and CatBoost-Cox address different prediction objectives, their results are interpreted separately rather than directly compared using the same performance metric. This work intends to provide an exhaustive set of qualitative and quantitative assessments of the model behavior in the face of industrial conditions.

4.1 Overall predictive performance across machine learning models

The first set of experiments investigates the predictive accuracies of the six machine-learning models using four evaluation metrics: RMSE, MAE, MedAE, and R^2. The reported results represent the mean performance obtained from the unseen outer-test partitions of the nested walk-forward validation procedure and are summarized in Table 7. The results indicate observable differences in metric-specific predictive behavior among the evaluated models rather than clear dominance of a single algorithm. Ensemble-based models generally achieved lower prediction errors, while the kernel-based and neural-network models remained competitive under the same evaluation framework.

Among the machine-learning models, Random Forest achieved the lowest mean RMSE (1.8813 days), MAE (1.0121 days), and MedAE (0.2552 days), indicating consistently low prediction errors across the evaluated metrics. CatBoost produced the highest mean R^2 value (0.6280), followed closely by GB (0.6237) and RF (0.6221). GB, SVR, CatBoost, and XGBoost also showed mean RMSE values close to that of RF, ranging from 1.8900 to 1.9170 days. MLP produced a higher mean RMSE of 2.2157 days and a lower mean R^2 of 0.3059, indicating comparatively weaker predictive performance under the same validation procedure.

The comparison with the reference

Table 7. . Predictive performance comparison of TTF models.

Model	RMSE (days)	MAE (days)	MedAE	R^2
RF	1.881	1.012	0.255	0.622
GB	1.890	1.049	0.356	0.624
XGBoost	1.917	1.039	0.278	0.600
SVR	1.892	1.135	0.469	0.575
MLP	2.216	1.343	0.580	0.306
CatBoost	1.900	1.067	0.364	0.628
Linear Regression	2.406	1.729	1.234	0.211
Temporal Mean	3.449	2.425	2.007	-0.122
Historical Median	3.479	2.456	2.200	-0.256
Historical Mean	4.125	3.458	3.506	-1.514
Previous Failure Interval	4.787	3.690	3.200	-0.996

approaches further demonstrates the value of the machine-learning models. All six machine-learning methods achieved lower mean RMSE and MAE values than the five reference approaches. Among the reference methods, Linear Regression showed the lowest mean RMSE at 2.4059 days, whereas the historical baselines produced considerably larger mean errors, with RMSE values ranging from 3.4489 to 4.7873 days. These results indicate that the predictive patterns captured by the machine-learning models could not be reproduced by simple historical averages, the most recent completed failure interval, or a conventional linear regression model.

Despite these differences, the mean RMSE values of RF, GB, SVR, CatBoost, and XGBoost were relatively close. Therefore, the numerical differences observed in Table 7 should not by themselves be interpreted as evidence that one machine-learning model is definitively superior to the others. As examined further in Section 4.6, pairwise statistical comparisons across the outer validation folds showed no statistically significant differences after adjustment for multiple comparisons. Accordingly, the results are better interpreted as reflecting differences in model-specific predictive behavior rather than a clear statistical dominance of a single algorithm.

4.2 Complementary CatBoost-Cox survival analysis

In addition to conventional CatBoost regression, CatBoost was evaluated using a Cox survival objective. Unlike the six direct regression models, CatBoost-Cox does not predict TTF directly in days. Instead, the model produces a relative failure-risk score, and its predictive discrimination was therefore assessed using Harrell's C-index. The overall CatBoost-Cox results are summarized separately in Table 8.

Table 8. Summary of CatBoost-Cox survival-model performance.

Measure	Result
Mean C-index	0.9155
SD of C-index	0.0336
95% bootstrap CI	0.8889–0.9410
Valid outer folds	5
Survival episodes	202
Observed events	201
Right-censored episodes	1

CatBoost-Cox achieved a mean C-index of 0.9155 with a standard deviation of 0.0336 across the five outer validation folds. Numerically, this represents a high level of concordance between predicted relative-risk ordering and observed survival ordering in the evaluated data. However, this result should not be compared numerically with the RMSE, MAE, MedAE, or R^2 values reported in Table 7 because these metrics represent fundamentally different prediction objectives. The CatBoost-Cox dataset contained 202 eligible survival episodes, of which 201 corresponded to observed failure events and one was right-censored. Therefore, although the Cox objective explicitly allows censored obser-

vations to contribute to model estimation, censoring was very limited in the present dataset. The resulting C-index should consequently be interpreted primarily as evidence of relative risk-ranking performance within the available episodes rather than as extensive validation under highly censored conditions.

4.3 Robustness and generalization under cross-validation

On the training data, nested cross-validation was used to evaluate the sensitivity of the models to data partitioning. Fig. 2 depicts the range of RMSE values of each of the compared ML models, revealing differences in their robustness and generalization performance obtained in different (nested) cross-validation splits. The RMSE results are shown in Table 9.

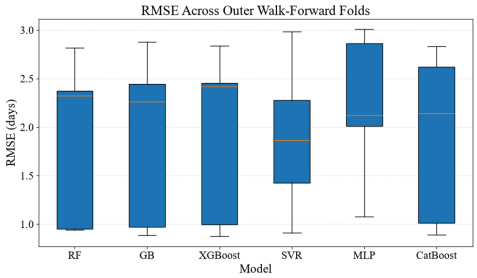


Fig. 2. RMSE distributions across the five outer walk-forward validation folds.

Table 9. Cross-validation RMSE statistics.

Model	Mean RMSE	Std.Dev.
RF	1.8813	0.8747
GB	1.8900	0.9053
XGBoost	1.9170	0.9100
SVR	1.8920	0.7937
MLP	2.2157	0.7737
CatBoost	1.9004	0.9018

As shown in Table 9, RF achieved the lowest mean RMSE (1.8813 days), followed closely by GB (1.8900 days), SVR (1.8920 days), CatBoost (1.9004 days), and XGBoost (1.9170 days). MLP exhibited the lowest across-fold variability (SD = 0.7737 days), followed by SVR (0.7937 days), but had the highest mean RMSE (2.2157 days). GB, CatBoost, and XGBoost showed slightly greater variability (SD = 0.9053, 0.9018, and 0.9100 days, respectively). RF achieved the lowest RMSE in Folds 1 and 5, while CatBoost, XGBoost, and SVR performed best in Folds 2, 3, and 4, respectively. Overall, no model consistently achieved the lowest error across all temporal folds, and differences in mean RMSE among RF, GB, SVR, CatBoost, and XGBoost were small.

To briefly illustrate the temporal generalization behavior of the models, Fig. 3 compares the observed and predicted TTF values obtained from RF and GB across the outer walk-forward test periods. These two models were selected for visualization because RF achieved the lowest mean RMSE, whereas GB achieved a closely comparable mean RMSE and represented a gradient-boosting approach for comparison with RF.

Both models generally followed the temporal changes in the observed TTF values, particularly during periods with relatively short TTF. However, larger prediction errors were observed at several high-TTF peaks, where both RF and SVR tended to underestimate the magnitude of the observed values. SVR also produced several slightly negative predictions during low-TTF periods, whereas RF predictions remained non-negative. These deviations indicate that prediction difficulty varied over time and became more pronounced during periods of relatively high TTF. Under the nested walk-forward validation framework, the 2021 observations were used only as training/history data; therefore, no outer-test predictions were available for that year.



(a) Random Forest



(b) SVR

Fig. 3. True vs. predicted TTF for Random Forest and SVR: nested cross-validation results.

The shaded regions from 2022 onward indicate the periods in which out-of-sample outer-test predictions were available. Fig. 3 therefore illustrates the temporal prediction behavior of RF and SVR across the available outer-test periods rather than demonstrating a clear advantage of one model over the other.

To further examine changes in predictive performance over time, the outer-test predictions were grouped by calendar year. This year-wise analysis was used as a descriptive assessment of temporal variation and was not treated as an additional

validation procedure. Because no outer-test predictions were available for 2021, the year-wise predictive evaluation was conducted only for the calendar years containing out-of-sample predictions.

Table 10 explores predictive performance across successive annual periods in which different descriptive MTBF levels were observed. The results reveal clear temporal variability across all models. In 2022, with an MTBF of 2.3459 days, RMSE values ranged from 1.8239 to 2.6732 days. CatBoost achieved the lowest RMSE during this period, whereas MLP showed the highest prediction error.

In 2023, prediction errors decreased across all models, with RMSE values ranging from 1.2508 to 1.4621 days, while the MTBF increased to 3.0500 days. RF achieved the lowest RMSE in this period. In 2024, prediction errors increased again, ranging from 2.6862 to 2.9430 days, although the MTBF remained similar at 3.0783 days. SVR achieved the lowest RMSE in 2024. The similar MTBF values observed in 2023 and 2024, despite substantially different RMSE levels, indicate that prediction error cannot be explained by MTBF alone.

Importantly, the relative performance of the models also changed across the annual periods. CatBoost achieved the lowest RMSE in 2022, RF in 2023, and SVR in 2024, indicating that no single model consistently achieved the lowest prediction error across all years. These findings demonstrate temporal variation in predictive behavior and show that aggregate cross-validation results may obscure performance differences occurring during specific operating periods. Therefore, time-structured evaluation is important when assessing TTF prediction models under changing manufacturing conditions.

Although nested walk-forward validation preserves the chronological structure of the data and evaluates the models on later unseen observations, it does not automatically adapt to future changes in operating conditions. Adaptive retraining and drift-detection approaches, such as sliding-window updating or ADWIN, may therefore be investigated in future work to support model stability under evolving manufacturing environments.

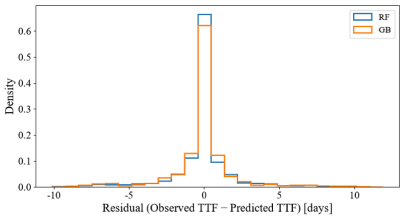
4.4 Error distribution and bias analysis

While overall predictive performance can be summarized using aggregate evaluation metrics, residual analysis provides additional insight into the direction and variability of prediction errors. Fig. 4 presents the residual distributions of the six machine-learning models. Most models showed residuals concentrated around zero, although extended tails were observed, indicating that relatively large prediction errors occurred for a smaller number of observations. Tree-based models generally exhibited residual distributions centered near zero, while SVR and MLP showed noticeable dispersion. These patterns suggest differences in error variability rather than strong evidence of a consistent systematic bias across the entire dataset. In practical maintenance applications, under-prediction may nevertheless be important because consistently conservative TTF estimates could lead to earlier maintenance intervention than necessary.

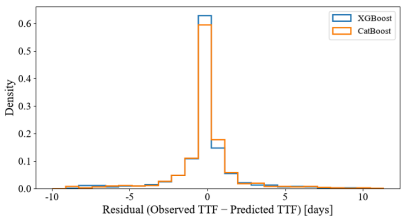
Most residuals were concentrated near zero, although extended tails were observed for all models. RF and GB showed relatively concentrated residual distributions, while XGBoost and CatBoost exhibited broadly similar patterns. SVR and MLP also showed residuals centered near zero but with noticeable dispersion.

Table 10. Year-wise test performance.

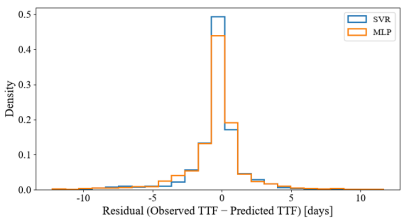
Year	RMSE on the method						MT BF
	RF	GB	XGBoost	SVR	MLP	CatBoost	
2022	1.9463	1.9182	2.362	2.0375	2.6732	1.8239	2.3459
2023	1.2508	1.3811	1.3324	1.3486	1.3358	1.4621	3.0500
2024	2.8936	2.9015	2.9024	2.6862	2.8161	2.9430	3.0783



(a) RF vs. GB



(b) XGBoost vs. CatBoost



(c) SVR vs. MLP

Fig. 4. Residual distributions of TTF prediction models.

Positive residuals indicate underprediction, whereas negative residuals indicate overprediction.

Both RF and GB exhibited relatively narrow and concentrated residual distributions, while XGBoost also showed a comparatively concentrated error pattern. Quantile-based error assessment showed that RF achieved the lowest median absolute error (MedAE = 0.2552 days), followed by XGBoost (MedAE = 0.2779 days). RF also achieved the lowest mean RMSE (1.8813 days) and MAE (1.0121 days). GB, SVR, CatBoost, and XGBoost produced similar mean RMSE values of 1.8900, 1.8920, 1.9004, and 1.9170 days, respectively. These results indicate differences in the central error and spread of the predictions rather than a clear advantage of a single model across all aspects of performance.

Negative predictions were also examined because negative TTF estimates are not physically meaningful in an operational maintenance context. RF produced no negative predictions, whereas GB, XGBoost, SVR, MLP, and CatBoost produced 154, 120, 163, 158, and 163 negative predictions, respectively. These values were retained in their original form and were not truncated to zero before calculating the primary performance metrics, thereby avoiding artificial improvement of the reported prediction accuracy. Their occurrence was instead reported separately as an operational limitation of the corresponding models.

In order to assess the potential financial impact of conservative TTF underprediction, the economic evaluation on an hourly basis was employed to appraise the potential cost associated with an unnecessarily early maintenance action. The analysis assumed an hourly production value of approximately 2.88 million THB (Thai Baht), representing the production value under full-capacity operation. Preventive maintenance costs included an average labor cost of 250 THB per hour (h), while rep-

resentative individual spare-part costs were 90,000 THB, 6,500 THB, and 1,200 THB, respectively. These figures were derived from plant operating information and prevailing market estimates and were applied as illustrative economic inputs to evaluate the potential consequences of conservative TTF underprediction. Preventive maintenance cost per hour is first given by

$$C_{PM} = C_{labor} + \frac{C_{spare}}{t_{PM}}, \qquad (4.1)$$

where C_{labor} is the hourly labor cost, C_{spare} is the spare-part cost per maintenance action while t_{PM} stands for the duration of the preventive maintenance in hours. The production value per hour, then, is expressed as

$$C_{throughput} = P \times R, \qquad (4.2)$$

where P is the selling price per unit and R is the production rate (units/h). By using Eqs. (4.1)-(4.2), the total hourly excess cost associated with an unnecessary maintenance intervention can be written as:

$$C_{Excess} = C_{throughput} + C_{PM}. \qquad (4.3)$$

Eq. (4.3) therefore combines the direct preventive maintenance cost with the opportunity cost associated with production interruption. The illustrative hourly economic impact of SVR underprediction is summarized in Table 11. The table presents three representative scenarios in which an underestimated TTF value could potentially trigger preventive maintenance earlier than necessary.

Across the three representative scenarios, the largest component of the estimated excess hourly cost was the opportunity cost associated with production interruption under full-capacity operation. In the high-value spare-component scenario, the combined effects of preventive maintenance and production interruption resulted in an estimated excess cost of approximately 2.93 million THB per hour. In the other cases, the estimated excess hourly costs remained above 2.88 million THB per hour, even though the spare-part costs in Cases 2 and 3 were substantially lower. Under the stated assumptions, this indicates that the production opportunity cost contributed more strongly to the total economic estimate than the direct maintenance-material cost.

The findings illustrate that TTF underprediction may have important operational consequences when model outputs are used to support maintenance timing decisions. Conservative TTF estimates could potentially lead to maintenance being initiated earlier than necessary, thereby increasing both direct maintenance costs and production opportunity losses. Consequently, underprediction behavior, particularly at relatively high TTF values, should be considered together with standard predictive-performance measures when evaluating predictive maintenance models for real-world decision support.

In operational terms, the estimated opportunity cost of approximately 2.9 million THB per hour illustrates the potential economic exposure associated with an unnecessary production interruption under full-capacity operation. Reducing TTF underprediction may therefore help reduce the likelihood of premature maintenance interventions and the associated opportunity losses. However, the actual economic benefit would depend on the duration of the avoided interruption, prevailing production conditions, and the maintenance decision rule applied in practice.

Table 11. Economic impact of SVR underprediction on preventive maintenance (hourly basis).

Case	Observed TTF (day)	SVR-Estimated TTF (day)	Spare Part Type	Spare Part Price (THB)	PM Cost (THB/h)	Excess Cost (THB/h)
1	22.00	15.36	A	90,000	54,250	2,934,250
2	26.00	15.07	B	1,200	1,358	2,881,358
3	28.00	14.73	C	6,500	3,829	2,883,829

4.5 Computational efficiency and practical deployability

Apart from predictive performance, we assessed computational efficiency as appropriate for real-world use. Table 12 displays average training time and prediction time per instance for models. All computational benchmarks were recorded using the hardware environment specified in Section 3.3 to ensure consistency and comparability across the models.

Table 12. Model efficiency comparison.

Model	Training Time (s)	Prediction Time (ms*)
SVR	2.664	0.011
XGBoost	3.091	0.004
GB	7.322	0.004
RF	8.595	0.009
CatBoost	12.029	0.004
MLP	25.868	0.010

* ms denotes milliseconds (10^{-3} seconds).

Among the six machine-learning models, SVR showed the shortest mean fitting time at 2.644 s, followed by XGBoost at 3.091 s and GB at 7.323 s. RF required 8.595 s, CatBoost required 12.030 s, and MLP showed the longest mean fitting time at 25.869 s. In terms of prediction efficiency, GB showed the lowest mean outer-test batch prediction time at 3.735 ms, followed closely by XGBoost at 4.064 ms and CatBoost at 4.280 ms. RF, MLP, and SVR required 9.291, 10.580, and 10.777 ms, respectively.

The computational results should be interpreted together with the predictive performance obtained under the same validation framework. As shown in Table 7, the mean RMSE values of the six machine-learning models ranged from 1.8813 to 2.2157 days. RF achieved the lowest mean RMSE (1.8813 days), MAE (1.0121 days), and MedAE (0.2552 days), whereas CatBoost achieved the highest mean R^2 (0.6280). GB, SVR, CatBoost, and XGBoost produced closely comparable mean RMSE values of 1.8900, 1.8920, 1.9004, and 1.9170 days, respectively.

Overall, the evaluated models exhibited different descriptive trade-offs between predictive and computational performance. RF produced the lowest mean prediction errors across RMSE, MAE, and MedAE, whereas SVR required the shortest mean fitting time. XGBoost combined relatively short fitting and prediction times with predictive performance close to that of the other tree-based models. GB achieved the shortest mean outer-test batch prediction time, while CatBoost produced the highest mean R^2. MLP required the longest mean fitting time and also produced the highest mean RMSE among the six machine-learning models.

Although differences were observed in the performance metrics of the evaluated models, their pairwise RMSE differences were further examined using the statistical procedure described in Section 4.6. Therefore, model selection for deployment should be guided by computational constraints, error characteristics, temporal sta-

bility, and operational priorities rather than assumptions of inherent algorithmic superiority.

According to the broader predictive-maintenance literature, previous studies have shown much larger operational improvements. For instance, Arowolo [20] observed a 35–45% reduction in unplanned downtime in energy-intensive industrial systems through predictive maintenance based on ML and IoT systems. This provides a contextual reference for the potential operational benefit of predictive maintenance but should not be interpreted as direct evidence that the same level of improvement would be achieved by the models evaluated in the present study.

4.6 Statistical significance

4.6.1 Statistical significance of model performance differences

Statistical significance tests were conducted based on paired RMSE differences across identical validation subsamples. Pairwise comparisons were performed using the Wilcoxon signed-rank test on the RMSE values obtained from the same five outer walk-forward validation folds. Since the analysis included six machine-learning models, a total of 15 pairwise comparisons were performed. To address the multiple-comparison, the raw p-values were adjusted using the Holm method to control the family-wise error rate. Statistical significance was assessed at $\alpha = 0.05$ based on the Holm-adjusted $p-$values. As indicated in Table 13, none of the pairwise differences were statistically significant after Holm correction.

Although differences in mean RMSE were observed among the six machine-learning models, none of the pairwise comparisons reached statistical significance after adjustment for multiple testing. The smallest raw $p-$value was observed for the comparison between SVR and MLP ($p = 0.0625$); after Holm correction, the corresponding adjusted $p-$value was 0.9375. Thus, the present analysis did not provide sufficient statistical evidence that the fold-level RMSE distributions differed between any pair of machine-learning models.

The models nevertheless exhibited different descriptive performance characteristics. RF achieved the lowest mean RMSE, MAE, and MedAE, whereas CatBoost achieved the highest mean R^2. SVR required the shortest search-and-refit time, while GB and XGBoost showed comparatively low prediction times. MLP exhibited the lowest fold-to-fold standard deviation of RMSE, although its mean RMSE was higher than those of the other machine-learning models. These differences should therefore be interpreted as descriptive characteristics of model behavior in the present dataset rather than as evidence of statistically significant superiority.

Interpretation of the statistical comparisons should also consider the limited number of paired evaluation units. Although the complete dataset comprised 1,560 operational records, the Wilcoxon comparisons were based on only five paired outer-fold RMSE values for each model. With such a small number of paired folds, the Wilcoxon signed-rank test yields relatively discrete p-values and has limited ability to detect small performance differences. Accordingly, failure to reject the null hypothesis should not be interpreted as evidence that the models have equivalent predictive performance. Instead, the statistical results indicate that the present validation design did not provide sufficient evidence to establish statistically significant pairwise differences.

The observed metric-specific differ-

Table 13. Pairwise Wilcoxon signed-rank tests with Holm correction for model performance differences on RMSE.

Model Comparison	Test Used	*p*-value	Holm-adjusted *p*-value	Significance
RF vs. GB	Wilcoxon signed-rank	0.8125	1.0000	No
RF vs. XGBoost	Wilcoxon signed-rank	0.3125	1.0000	No
RF vs. SVR	Wilcoxon signed-rank	1.0000	1.0000	No
RF vs. MLP	Wilcoxon signed-rank	0.3125	1.0000	No
RF vs. CatBoost	Wilcoxon signed-rank	0.8125	1.0000	No
GB vs. XGBoost	Wilcoxon signed-rank	0.6250	1.0000	No
GB vs. SVR	Wilcoxon signed-rank	0.6250	1.0000	No
GB vs. MLP	Wilcoxon signed-rank	0.3125	1.0000	No
GB vs. CatBoost	Wilcoxon signed-rank	1.0000	1.0000	No
XGBoost vs. SVR	Wilcoxon signed-rank	1.0000	1.0000	No
XGBoost vs. MLP	Wilcoxon signed-rank	0.4375	1.0000	No
XGBoost vs. CatBoost	Wilcoxon signed-rank	0.8125	1.0000	No
SVR vs. MLP	Wilcoxon signed-rank	0.0625	0.9375	No
SVR vs. CatBoost	Wilcoxon signed-rank	0.6250	1.0000	No
MLP vs. CatBoost	Wilcoxon signed-rank	0.3125	1.0000	No

ences are therefore more appropriately interpreted descriptively, since the hypothesis tests did not provide sufficient evidence of statistically significant pairwise differences among the models.

4.6.2 Statistical assessment of catboost-cox survival performance

Because CatBoost-Cox was evaluated using C-index rather than RMSE, it was not included in the 15 pairwise RMSE comparisons presented in Table 14. Statistical uncertainty for CatBoost-Cox was instead evaluated separately using the five outer-fold C-index estimates.

Bootstrap resampling of the five fold-level C-index estimates produced a 95% confidence interval of 0.8889–0.9410 around the mean C-index. An additional exploratory one-sided Wilcoxon signed-rank test compared the five fold-level C-index values with the non-discriminatory reference value of 0.50 and produced $p = 0.03125$.

The Wilcoxon result should nevertheless be interpreted cautiously. Only five outer-fold C-index values were available, and the expanding training windows overlapped temporally. Therefore, the fold-level observations cannot be regarded as fully independent experimental replications. For this reason, the mean C-index, standard deviation, and confidence interval were emphasized as descriptive measures, whereas the one-sample Wilcoxon result was treated as supplementary exploratory evidence.

Table 14. Statistical assessment of CatBoost-Cox.

Measure	Result
Mean C-index	0.9155
SD C-index	0.0336
Bootstrap 95% CI	0.8889-0.9410
Reference C-index	0.5000
Exploratory Wilcoxon *p*-value	0.03125
Valid outer folds	5

The survival analysis was additionally characterized by very limited censoring, with 201 observed-event episodes and only one right-censored episode. Accordingly, the current results provide stronger information regarding failure-risk ranking than regarding performance in datasets with substantial right censoring. To further sup-

Table 15. Decision guidelines for model selection in TTF prediction.

Operational Priority	Recommended Model	Rationale
Minimize overall prediction error	RF	Lowest mean RMSE (1.8813 days) and mean MAE (1.0121 days)
Maximize overall goodness of fit	CatBoost	Highest mean (0.6280)
Maintain consistent prediction performance across validation periods	MLP	Lowest RMSE standard deviation (0.7737 days), although its mean RMSE was higher
Minimize median prediction error	RF	Lowest mean MedAE (0.2552 days), followed by XGBoost (0.2779 days)
Minimize model-fitting time	SVR	Shortest mean search-and-refit time (2.9996 s)
Minimize prediction time	GB	Lowest mean outer-test batch prediction time (3.735ms), closely followed by XGBoost (4.064 ms)
Avoid negative TTF predictions	RF	No negative TTF predictions were produced, whereas negative estimates occurred for the other evaluated models
Rank relative failure risk while incorporating censoring	CatBoost-Cox	Survival formulation evaluated with mean C-index = 0.9155

port practical interpretation, Table 15 summarizes the descriptive characteristics of the evaluated models according to different operational priorities. These guidelines are intended to assist maintenance engineers in relating model characteristics to deployment requirements and should not be interpreted as evidence of statistically significant superiority.

Model selection should therefore be considered from a practical deployment perspective rather than relying solely on statistical rankings. Based on the descriptive results, RF exhibited the lowest overall prediction errors and did not produce negative TTF estimates, whereas CatBoost achieved the highest mean R^2. SVR required the shortest search-and-refit time, while GB and XGBoost provided comparatively low outer-test batch prediction times. MLP showed the lowest RMSE variability across the five temporal folds, although this lower variability was accompanied by a higher mean RMSE. These characteristics represent descriptive trade-offs rather than statistically significant differences among the algorithms. None of the 15 pairwise RMSE comparisons reached statistical significance after Holm correction. Future work may therefore investigate model robustness using additional independent temporal validation periods or external datasets, which would provide more paired evaluation units and greater statistical power for detecting potentially small performance differences. Ensemble strategies combining complementary characteristics of the individual models may also be investigated.

4.7 Sensitivity analysis of data preprocessing choices

In addition to the primary hyperparameter-tuning procedure, an extensive sensitivity analysis was carried out to examine the effects of exploring broader or alternative model configurations on predictive robustness and computational

trade-offs. For the tree-based ensemble models, the number of estimators or boosting iterations was extended beyond the initial search ranges, reaching up to 600 estimators or iterations in some scenarios. For SVR, alternative kernel functions were systematically investigated, while different hidden-layer architectures were examined for MLP. All sensitivity scenarios were evaluated under the same nested walk-forward validation framework used in the primary analysis. The predictive performance of the five reference baseline models is reported in Table 7 for comparison. The results of the extended model-configuration analysis are presented in Table 16.

For the tree-based ensemble models, RF, GB, XGBoost, and CatBoost showed relatively limited changes in predictive performance across the expanded configurations. RF produced mean RMSE values ranging from 1.8451 to 1.8476 days, compared with 1.8813 days under the primary configuration. GB ranged from 1.8391 to 1.8624 days, compared with the primary value of 1.8900 days, while XGBoost ranged from 1.8567 to 1.8950 days compared with 1.9170 days in the primary analysis. CatBoost similarly produced mean RMSE values between 1.8952 and 1.9144 days, compared with 1.9004 days under the primary configuration. Although some expanded configurations produced slightly lower RMSE values, increasing the ensemble size did not result in a consistent improvement in predictive accuracy. In contrast, computational cost increased more clearly as model complexity increased. For example, the search-and-refit time for RF increased from 12.840 s with 100–300 estimators to 33.925 s with 500–600 estimators. Similarly, GB increased from 10.843 to 28.491 s, XGBoost from 4.325 to 9.395 s, and CatBoost from 9.051 to 23.585 s across the evaluated ranges. These results indicate that expanding ensemble size provided only modest changes in predictive performance while increasing computational requirements

In SVR, kernel selection had a substantially greater influence on performance than the moderate expansion of ensemble size observed for the tree-based models. The polynomial, linear, and sigmoid kernels produced mean RMSE values of 2.9659, 2.2696, and 2.5085 days, respectively, with corresponding mean R^2 values of 0.0246, 0.3811, and 0.1765. All three alternative kernels performed less favorably than the primary SVR configuration, which achieved a mean RMSE of 1.8920 days. The polynomial kernel was particularly computationally demanding, requiring a mean search-and-refit time of 10,121.526 s, compared with 4.539 s for the linear kernel and 1.632 s for the sigmoid kernel. These results indicate that SVR was highly sensitive to kernel selection and that greater computational complexity did not necessarily lead to improved predictive performance. The MLP sensitivity scenarios also showed that deeper or wider architectures did not guarantee consistent performance gains. The single-hidden-layer configuration produced a mean RMSE of 2.3566 days, while the two-hidden-layer configuration achieved 2.1930 days. The deeper three- to four-hidden-layer configuration further reduced the mean RMSE to 2.0432 days, compared with 2.2157 days for the primary MLP configuration. The corresponding mean R^2 values were 0.0943, 0.3713, and 0.5216, respectively. However, the computational requirements varied considerably, with mean search-and-refit times of 28.016, 121.123, and 54.226 s for Scenarios 1–3.

In addition, convergence warnings were observed in some MLP fits, suggesting that optimization stability remained sensitive to network architecture. Therefore, although the deeper configuration produced a lower RMSE, this result should be interpreted as evidence of architectural sensitivity rather than as justification for retrospectively replacing the primary MLP configuration.

Overall, the sensitivity analysis indicates that expanding hyperparameter ranges or model complexity did not produce consistent accuracy improvements across all models. The tree-based ensemble models showed relatively stable predictive performance over the evaluated configurations, while their computational cost generally increased with greater model complexity. In contrast, SVR and MLP were more sensitive to structural choices, particularly kernel type and hidden-layer architecture. These findings complement the statistical comparison presented in Section 4.6 by showing that moderate changes in model configuration did not substantially alter the general comparative behavior of the tree-based models.

Together with the preprocessing sensitivity analysis, the results suggest that model robustness should be assessed not only from predictive accuracy but also from computational efficiency, optimization stability, and consistency across temporally separated validation folds.

4.8 Sensitivity analysis of alternative preprocessing configurations

A sensitivity analysis was conducted to examine whether predictive performance was affected by alternative treatments of missing values and extreme observations. Four preprocessing configurations were evaluated under the same nested walk-forward validation framework: median imputation without clipping, mean imputation without clipping, median imputation with IQR-based clipping, and mean imputation with IQR-based clipping. Median imputation without clipping was retained as the prespecified primary configuration, whereas the remaining configurations were used only to assess the robustness of the results. All preprocessing operations were fitted using the training data within each fold to prevent information leakage.

The results are summarized in Table 18. RF, GB, XGBoost, and CatBoost showed relatively small changes in mean RMSE across the four preprocessing configurations. For RF, the mean RMSE ranged from 1.8344 to 1.8813 days, while GB ranged from 1.8438 to 1.8900 days. XGBoost showed mean RMSE values between 1.8414 and 1.9170 days, whereas CatBoost ranged from 1.8642 to 1.9232 days. These relatively narrow ranges indicate that the predictive performance of the tree-based ensemble models was generally stable under the evaluated preprocessing alternatives.

In contrast, SVR showed greater sensitivity to the treatment of extreme observations. Its mean RMSE was 1.8920 days under the primary median-imputation-without-clipping configuration but increased to 2.0776 and 2.0718 days when IQR-based clipping was applied with median and mean imputation, respectively. This corresponds to an increase of approximately 9–10% relative to the primary configuration. The results suggest that SVR was more strongly affected by modifications to the distribution of extreme observations than the tree-based ensemble models.

MLP also showed noticeable variation across the preprocessing con-

Table 16. Sensitivity analysis of alternative model configurations.

Scenario	Model	Key parameters	RMSE (days)	MAE (days)	MedAE (days)	R^2	Training Time (s)
1	RF	n_estimators: 100–300	1.8476	0.9992	0.2415	0.6356	12.840
2	RF	n_estimators: 300–500	1.8451	0.9978	0.2457	0.6365	24.403
3	RF	n_estimators: 500–600	1.8458	0.9990	0.2462	0.6351	33.925
1	GB	n_estimators: 100–300	1.8587	1.0409	0.3479	0.6414	10.843
2	GB	n_estimators: 300–500	1.8391	1.0250	0.3365	0.6495	20.696
3	GB	n_estimators: 500–600	1.8624	1.0462	0.3468	0.6462	28.491
1	XGBoost	n_estimators: 100–300	1.8567	1.0439	0.3529	0.6360	4.325
2	XGBoost	n_estimators: 300–500	1.8950	1.0645	0.3467	0.6357	7.186
3	XGBoost	n_estimators: 500–600	1.8892	1.0648	0.3692	0.6350	9.395
1	SVR	kernel=poly	2.9659	2.1146	1.4907	0.0246	10121.526
2	SVR	kernel=linear	2.2696	1.4884	0.8103	0.3811	4.539
3	SVR	kernel=sigmoid	2.5085	1.7171	1.1010	0.1765	1.632
1	MLP	hidden layers (64 to 128), (256 to 512)	2.3566	1.4970	0.7667	0.0943	28.016
2	MLP	hidden layers (128 to 64), (192 to 96), (256 to 128)	2.1930	1.3609	0.6653	0.3713	121.123
3	MLP	hidden layers (128-64-32), (192-96-48), (256-128-64), (128-64-32-16)	2.0432	1.1591	0.4365	0.5216	54.226
1	CatBoost	iterations 100–300	1.9039	1.0665	0.3350	0.6243	9.051
2	CatBoost	iterations 300–500	1.9144	1.0714	0.3430	0.6269	17.099
3	CatBoost	iterations 500–600	1.8952	1.0580	0.3170	0.6395	23.585

Table 17. Sensitivity analysis of alternative preprocessing configurations based on mean RMSE across the outer validation folds.

Preprocessing Scenario	RF	GB	XGBoost	SVR	MLP	CatBoost
Median imputation, no clipping (Primary)	1.8813	1.8900	1.9170	1.8920	2.2157	1.9004
Mean imputation, no clipping	1.8561	1.8760	1.9075	1.9032	2.0616	1.9232
Median imputation, IQR clipping	1.8525	1.8671	1.8414	2.0776	2.1377	1.8642
Mean imputation, IQR clipping	1.8344	1.8438	1.8843	2.0718	2.1214	1.8773

figurations, with mean RMSE values ranging from 2.0616 to 2.2157 days. Under the mean-imputation-with-IQR-clipping configuration, one MLP fit reached the maximum of 1,000 iterations without convergence. Therefore, the MLP sensitivity results should be interpreted with some caution because its optimization behavior was less stable under certain preprocessing conditions.

Overall, the preprocessing sensitivity analysis indicates that the principal conclusions for RF, GB, XGBoost, and CatBoost were relatively robust to reasonable alternative preprocessing choices, whereas SVR and MLP showed greater sensitivity. Importantly, the alternative preprocessing configurations were not used to select or replace the primary analysis according to outer-test performance. Instead, they were evaluated solely as robustness checks to determine whether the reported findings depended strongly on a particular treatment of missing values or extreme observations.

5. Conclusion

The paper presents a systematic benchmarking framework for Time-to-

Failure (TTF) prediction in an industrial beverage-packaging labeling machine operating under temporally varying production conditions. The analysis used four years of operational and maintenance records comprising 1,560 observations and 479 documented unplanned failure events. Six machine-learning regression models, including Random Forest (RF), Gradient Boosting (GB), XGBoost, Support Vector Regression (SVR), Multi-Layer Perceptron (MLP), and CatBoost, were evaluated together with five reference approaches using nested walk-forward validation. In addition, CatBoost was implemented with a Cox survival objective (CatBoost-Cox) as a complementary censor-aware approach for relative failure-risk assessment.

The results showed that all six machine-learning regression models achieved lower mean RMSE values than the reference approaches. RF achieved the lowest mean RMSE (1.8813 days), MAE (1.0121 days), and MedAE (0.2552 days), whereas CatBoost achieved the highest mean R^2 (0.6280). However, pairwise Wilcoxon signed-rank tests across the five outer validation folds showed no statistically significant RMSE differences after Holm correction. The smallest raw $p-$value was obtained for SVR versus MLP ($p = 0.0625$), which increased to 0.9375 after adjustment. Therefore, the observed differences should be interpreted as descriptive variations in model behavior rather than evidence of statistically significant superiority.

The complementary CatBoost-Cox analysis achieved a mean C-index of 0.9155 (SD = 0.0336), with a 95% bootstrap confidence interval of 0.8889–0.9410. These results indicate a high level of concordance in relative failure-risk ordering across the evaluated temporal folds. However, the survival analysis included 202 episodes with only one right-censored observation; therefore, the findings primarily reflect risk-discrimination performance within the available data rather than performance under substantial censoring.

The temporal, residual, and computational analyses further showed that model behavior varied across operating periods and evaluation criteria. RF produced no negative TTF predictions, whereas negative estimates occurred for GB, XGBoost, SVR, MLP, and conventional CatBoost. SVR required the shortest mean search-and-refit time, while MLP produced the shortest mean outer-test batch prediction time. The economic assessment also illustrated that TTF underprediction may result in substantial maintenance and production-opportunity costs, although these estimates represent illustrative scenarios rather than realized financial savings.

The sensitivity analyses indicated that the tree-based regression models were relatively stable across the evaluated model and preprocessing configurations, whereas SVR and MLP showed greater sensitivity to modeling choices. CatBoost-Cox also showed limited variation across the evaluated preprocessing configurations, with mean C-index values ranging from approximately 0.9155 to 0.9219.

Several limitations should be acknowledged. The study examined a single industrial labeling machine, relied on five outer validation folds, and included only one right-censored episode in the CatBoost–Cox analysis. Future research should evaluate the framework across additional machines, external datasets, and other TTF prediction problems to assess generalizability and robustness. Methodological extensions could investigate LightGBM models, other modern machine learn-

ing models, and alternative survival models, including Weibull accelerated failure-time models and random survival forests.

Overall, the study demonstrates the potential of machine-learning methods for predictive maintenance. The six regression models provide direct TTF estimates, while conventional Cox regression and the proposed CatBoost–Cox integration offer complementary assessments of relative failure risk. The findings support a temporally validated, application-specific framework that integrates TTF prediction and survival analysis, while highlighting the need for broader external validation.

Acknowledgements

The data used in this work is supported by Singha Beverage Co., Ltd. This study was conducted under ethical approval from the Human Research Ethics Committee of Prince of Songkla University (PSU-HREC-2025-086-1-3).

References

[1] Megdadi E, Mohamed A, Shaalan K. Machine learning-driven best-worst method for predictive maintenance in Industry 4.0. Automation. 2025;6(4):91.

[2] Florian E, Sgarbossa F, Zennaro I. Machine learning-based predictive maintenance: a cost-oriented model for implementation. Int J Prod Econ. 2021;236:108114.

[3] Justus V, Kanagachidambaresan GR. Machine learning based fault-oriented predictive maintenance in Industry 4.0. Int J Syst Assur Eng Manag. 2024;15:462-74.

[4] Ayvaz S, Alpay K. Predictive maintenance system for production lines in manufacturing: a machine learning approach using IoT data in real-time. Expert Syst Appl. 2021;173:114598.

[5] Akyaz T, Engin D. Machine learning-based predictive maintenance system for artificial yarn machines. IEEE Access. 2024;12:125446-125461.

[6] Mohammed NA, Abdulateef OF, Hamad AH. An IoT and machine learning-based predictive maintenance system for electrical motors. J Eur Syst Autom. 2023;56(4):651-6.

[7] Rosati R, Romeo L, Cecchini G, Tonetto F, Viti P, Mancini A, et al. From knowledge-based to big data analytic model: a novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0. J Intell Manuf. 2023;34:107-21.

[8] Drakaki M, Karnavas YL, Tziafettas IA, Linardos V, Tzionas P. Machine learning and deep learning based methods toward Industry 4.0 predictive maintenance in induction motors: state of the art survey. J Ind Eng Manag. 2022;15(1):31-57.

[9] Zonta T, da Costa CA, da Rosa Righi R, de Lima MJ, da Trindade ES, Li GP. Predictive maintenance in the Industry 4.0: a systematic literature review. Comput Ind Eng. 2020;150:106889.

[10] Abidi MH, Mohammed MK, Alkhalefah H. Predictive maintenance planning for Industry 4.0 using machine learning for sustainable manufacturing. Sustainability. 2022;14(6):3387.

[11] Çınar ZM, Abdussalam Nuhu A, Zeeshan Q, Korhan O, Asmael M, Safaei B. Machine learning in predictive maintenance towards sustainable smart manufacturing in Industry 4.0. Sustainability. 2020;12(19):8211.

[12] Achouch M, Dimitrova M, Ziane K, Sattarpanah Karganroudi S, Dhouib R, Ibrahim H, et al. On predictive maintenance in Industry 4.0: overview, models, and challenges. Appl Sci. 2022;12(16):8081.

[13] Aminzadeh A, Sattarpanah Karganroudi S, Majidi S, Dabompre C, Azaiez K, Mitride C, et al. A machine learning implementation to predictive maintenance and monitoring of industrial compressors. Sensors. 2025;25(4):1006.

[14] Yadav DK, Kaushik A, Yadav N. Predicting machine failures using machine learning and deep learning algorithms. SMSE. 2024;3:100029.

[15] Taşcı B, Omar A, Ayvaz S. Remaining useful lifetime prediction for predictive maintenance in manufacturing. Comput Ind Eng. 2023;184:109566.

[16] Jayanthi LN, Mohankumar N, Meenakshi R, Kathaian R, Subba Ramaiah V, Subramanian S. CatBoost-based predictive maintenance for industrial milling tool life and failure detection. In: Proceedings of the 12th International Conference on Communication and Signal Processing (ICCSP); 2026; Tamilnadu, India. p. 636-41.

[17] Thijssens OWM, Verhagen WJC. Application of extended Cox regression model to time-on-wing data of aircraft repairables. Reliab Eng Syst Saf. 2020;204:107136.

[18] Varalakshmi K, Kumar J. Optimized predictive maintenance for streaming data in industrial IoT networks using deep reinforcement learning and ensemble techniques. Sci Rep. 2025;15:27201.

[19] García FJÁ, Salgado DR. An approach for predictive maintenance decisions for components of an industrial multistage machine that fail before their MTTF: a case study. Systems. 2022;10(5):175.

[20] Arowolo M. Predictive maintenance of energy-intensive industrial equipment using IoT and machine learning technologies. IOSR J Mech Civ Eng. 2025;22(3):14-26.