

Article

Alarm Prediction in Predictive Maintenance: A Comparative Analysis of Temporal Windows for Machine Learning Models in Industrial Systems

Mario Caterino ^{1,*} , Tammamo Ucciero ¹ , Riccardo Emanuele Landi ², Luca Mozzillo ², Antonio Negro ², Federico Saponaro ³, Domenico Soriano ², Ivo Surano ³, Salvatore Miranda ⁴ , Roberto Macchiaroli ¹ and Marcello Fera ¹ 

¹ Department of Engineering, University of Campania “Luigi Vanvitelli”, 81100 Caserta, Italy; tammamo.ucciero3@studenti.unicampania.it (T.U.); roberto.macchiaroli@unicampania.it (R.M.); marcello.fera@unicampania.it (M.F.)

² Mare Engineering Group S.p.A., 80038 Pomigliano d’Arco, Italy; riccardo.landi@maregroup.it (R.E.L.); luca.mozzillo@maregroup.it (L.M.); antonio.negro@maregroup.it (A.N.); domenico.soriano@maregroup.it (D.S.)

³ Gelesis Srl, 73021 Calimera, Italy; fsaponaro@gelesis.com (F.S.); isurano@gelesis.com (I.S.)

⁴ Department of Industrial Engineering, University of Salerno, 84084 Fisciano, Italy; smiranda@unisa.it

* Correspondence: mario.caterino@unicampania.it; Tel.: +39 081-5013672

Abstract

Predictive maintenance (PdM) plays a key role in improving the reliability and efficiency of industrial systems by anticipating abnormal operating conditions through data-driven approaches. In this context, many PdM solutions rely on temporal windowing strategies applied to multivariate time series; however, the choice of observation and prediction horizons is often fixed a priori and rarely analyzed in a systematic way. This paper investigates the impact of temporal window design by formulating alarm prediction as a window-based classification problem, explicitly distinguishing between the observation horizon and the prediction horizon. A comprehensive experimental study is conducted on three real-world industrial datasets characterized by different process dynamics. Multiple machine learning and deep learning models are evaluated across a wide range of combinations of observation and prediction horizons. The results show that predictive performance is strongly influenced by the joint configuration of observation and prediction horizons, rather than by their individual selection. While increasing the prediction horizon generally reduces accuracy, the magnitude of this effect depends on system dynamics. Similarly, longer observation horizons substantially improve performance in processes with slow dynamics, while providing limited benefits in faster-evolving systems. Overall, the study highlights the importance of explicitly modeling temporal window parameters in predictive maintenance design, providing both methodological insights and practical guidelines for selecting appropriate temporal configurations in industrial alarm prediction tasks.

Keywords: predictive maintenance; artificial intelligence; temporal windowing; alarm prediction; industrial systems



Academic Editor: Mohd Shareduwan Mohd Kasihmuddin

Received: 9 April 2026

Revised: 16 May 2026

Accepted: 18 May 2026

Published: 25 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In today’s competitive industrial landscape, data-driven decision making is increasingly supported by artificial intelligence (AI). Among the concrete applications enabled by Industry 4.0/5.0, such as cloud manufacturing [1], human–robot interaction [2], and intelligent

logistics [3], predictive maintenance (PdM) has emerged as a key lever to reduce downtime and increase productivity [4], reduce maintenance costs [5] and energy consumption [6].

PdM can use historical data and real-time measurements to estimate the future state of machinery and equipment, enabling the prevention of failures [7]. Its implementation is based on an integrated technological ecosystem that includes IoT (Internet of Things) sensors for real-time condition monitoring, reliable network infrastructures for continuous data transmission, edge computing solutions for the immediate processing of streams coming from sensors, big data platforms for data storage, and artificial intelligence algorithms for building predictive models.

Nevertheless, large-scale deployment is still challenged by several major issues: implementing technologies such as IoT sensors and edge computing solutions may result in high costs [8]. Data quality [9] and governance are often major issues, limiting the implementation of PdM policies. In addition, deploying PdM infrastructures requires specialized expertise that is still not widely available. Lastly, implementing PdM may also require additional investment in cybersecurity, since data transmitted to the cloud can be more vulnerable to cyberattacks [10].

Recent predictive-maintenance literature confirms that data-driven PdM is moving from isolated fault-diagnosis models toward deployment-oriented pipelines in which data quality, class imbalance, model interpretability, and real-world validation are central concerns [11]. Recent reviews of ML-driven PdM across industrial domains show that sensor heterogeneity and imbalanced fault labels remain major obstacles to generalizable predictive models, while deep-learning surveys emphasize that the suitability of CNN, recurrent, attention-based, and hybrid architectures depends strongly on the temporal structure and quality of the available data [12].

Within this broader context, temporal framing is a particularly important but still underspecified design choice. Industrial alarm prediction usually requires converting continuous multivariate time series into observation windows and future prediction windows. Existing studies on sliding-window failure prediction have shown that the choice of lookback and prediction horizons can substantially affect performance and that the optimal configuration is often application-dependent [13,14]. However, temporal windows are still frequently treated as a preprocessing choice rather than as a modeling variable with theoretical and operational implications.

Despite these challenges, interest in PdM continues to grow in both academic and industrial settings. In industry, PdM is generally adopted to reduce unexpected machine failures and enhance the overall efficiency of production systems. In general, the data used for PdM are typically represented as multivariate time series collected from sensors and control systems [15]. The large volume of data and the non-trivial dependencies among variables motivate the use of AI systems, such as machine learning (ML) and deep learning (DL), which can exploit complex patterns to anticipate alarms [16]. In this context, formulating PdM as a window-based classification problem provides a practical operational framing: given a current window of observations, predict whether an alarm will occur in a future window. This approach yields actionable lead time while accommodating processes with different dynamics. Despite the growing body of literature addressing alarm and failure prediction in industrial systems, most existing studies assume fixed or implicitly defined temporal horizons. In particular, the joint impact of the observation window length and the prediction horizon on model performance is rarely investigated in a systematic manner [13,14].

The objective of this paper is therefore to investigate how the joint choice of observation horizon and prediction horizon influences the ability of different ML and DL models to anticipate industrial alarms from multivariate time-series data. The theoretical contribution

of the study is to frame temporal windowing as a coupled design problem in predictive maintenance: the observation window determines the historical evidence available to the classifier, the future window determines the actionable lead time, and their interaction is conditioned by the dynamic behavior of the monitored process. Accordingly, the study contributes by: systematically varying observation window and prediction window rather than fixing them a priori; comparing classical and deep learning models under the same temporal configurations; and deriving practical guidance for selecting temporal windows according to process dynamics, prediction reliability, and computational cost.

The remainder of the paper is structured as follows: Section 2 reviews the related work; Section 3 presents the problem and the methodology adopted in this study; Section 4 describes the experimental setup; Section 5 reports the results of the experimental study, which are discussed in Section 6; and finally, Section 7 concludes the paper.

2. Related Work

Predictive maintenance has been widely investigated as a data-driven strategy to improve the reliability and availability of industrial systems by anticipating abnormal operating conditions before the occurrence of failures [17]. In industrial environments, PdM typically relies on multivariate time series acquired from sensors, control systems, and programmable logic controllers, which continuously monitor the operational state of machinery and processes.

Depending on the application and data availability, predictive maintenance problems have been formulated using different paradigms. A significant portion of the literature focuses on failure time estimation or remaining useful life (RUL) prediction through regression-based approaches, in which degradation trajectories extracted from sensor data are modeled and extrapolated over time. Representative examples include data-driven methods that derive degradation indicators from vibration or process signals and estimate RUL using dynamic regression models [18]. Several studies further enrich this formulation by integrating diagnostic and prognostic stages within a unified framework. In particular, hybrid approaches combining health state estimation and regression-based RUL prediction have been proposed, for instance by modeling discrete degradation states through probabilistic models such as Hidden Markov Models and subsequently estimating RUL using polynomial regression [19]. More recent works leverage machine learning and deep learning techniques to improve RUL estimation accuracy and robustness. Neural-network-based models, including multilayer perceptrons, have been applied to production line equipment and benchmark datasets, demonstrating the effectiveness of data-driven regression approaches for modeling degradation trajectories and predicting time to failure [20]. In parallel, anomaly-aware RUL estimation frameworks have been introduced, where degradation onset is detected through statistical or learning-based monitoring and RUL prediction is activated only after abnormal behavior is identified, while still preserving a regression-based prognostic formulation [21].

PdM has also been widely formulated as an alarm or event prediction problem, where the objective is to anticipate the occurrence of abnormal conditions within a future time interval. This formulation is particularly common in industrial contexts characterized by discrete alarm mechanisms and complex process dynamics, where early warning enables timely operational or maintenance actions. Within this paradigm, PdM is typically formulated as a supervised classification problem based on fixed-length windows extracted from multivariate time series. Historical data are segmented into temporal windows and labeled according to the presence of alarms, faults, or abnormal events, allowing machine learning models to discriminate between normal and pre-alarm operating conditions. Classical classifiers as well as deep learning models have been used to capture nonlinear relation-

ships and temporal dependencies among process variables [22]. Several works further extend classification-based PdM by integrating anomaly detection and decision-making mechanisms. Hybrid approaches combining unsupervised anomaly detection with supervised classification have been proposed to generate alarm labels in scenarios where annotated data are scarce, enabling the prediction of abnormal events in real time [22]. Other studies exploit reinforcement learning or hybrid learning frameworks to support proactive or alarm-driven maintenance policies, where system states inferred from sensor data guide maintenance-related decisions [23].

Regardless of the specific predictive maintenance formulation adopted, the adoption of temporal windowing strategies is a widespread practice in predictive maintenance applications dealing with industrial multivariate time series. Whether the problem is formulated as failure-time estimation, RUL prediction, or alarm classification, raw sensor signals are commonly segmented into fixed-length temporal windows that serve as inputs to predictive models [13]. This approach enables the extraction of temporal patterns and correlations while providing a structured representation of continuous data streams. Within this framework, two temporal parameters play a central role: the observation horizon, which defines the length of historical data used to characterize the current system state, and the prediction horizon, which specifies the future time interval over which failures, alarms, or abnormal events are to be anticipated [24]. In regression-based formulations, the prediction horizon is often implicit and linked to the estimation of time-to-failure or degradation trajectories, whereas in classification-based approaches it is typically defined explicitly as a future window in which the occurrence of an event determines the class label [25].

The choice of observation and prediction horizons has important implications from both a modeling and an operational perspective. Longer observation windows may capture richer temporal dynamics and slow degradation trends, while shorter windows can favor responsiveness and reduce computational complexity [26]. Similarly, longer prediction horizons provide earlier warnings but may increase uncertainty and reduce predictive accuracy, whereas shorter horizons tend to yield more reliable predictions at the cost of reduced lead time [27]. These trade-offs highlight the strong coupling between temporal window design and the practical usefulness of predictive maintenance solutions. Despite their importance, temporal horizon parameters are frequently selected using heuristic or empirical criteria, such as domain expertise, trial-and-error experimentation, or constraints imposed by data availability and computational resources [28–30]. In many studies, window lengths are fixed a priori and remain unchanged across different models, datasets, or operating conditions, with limited discussion of their influence on predictive performance [31,32]. As a result, temporal windowing is often treated as a preprocessing step rather than as a key component of the predictive modeling process. Nevertheless, several studies have shown that the choice of temporal horizon parameters can substantially affect PdM and time-series forecasting performance [13,14,33]. Zhang et al. [26] showed that longer observation horizons can improve predictive accuracy, even for longer prediction horizons, as a larger historical context provides richer and more informative temporal patterns. However, using longer observation horizons also increases computational requirements and training time [34]. Therefore, an appropriate trade-off must be found between prediction accuracy and computational cost. In this respect, the choice of the learning algorithm also plays an important role. Elastic-Net Regularized Logistic Regression (ENRLR), which is a generalized linear model, has achieved good predictive performance with limited computational training costs, with training often completed in minutes [35]. Light Gradient Boosting Machine (LGBM) has also shown high predictive accuracy with relatively low training costs [36], although hyperparameter tuning can be challenging [37]. Polynomial Kernel Support Vector Machines (PK-SVM) have also been widely applied in PdM applications.

Several studies have demonstrated their effectiveness in achieving good predictive performance [38]; however, optimizing the kernel parameters to reduce computational cost typically requires an additional optimization step [39,40]. Hybrid Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) models are also frequently used in PdM [41]. In these architectures, the CNN component is typically used to extract high-level features and short-term dependency patterns from the data, while the LSTM component captures relationships among several time-series variables and models complex long-term temporal dependencies, leading to strong performance in both alarm prediction [42] and RUL estimation [43]. However, CNN-LSTM models may require substantial computational resources when processing large volumes of high-dimensional, heterogeneous, and noisy data [44]. Transformer-based Time-Series Classifiers (TBTSC) offer modeling advantages for long-range dependencies, although their performance may depend strongly on the quantity and quality of the available data [45].

ENRLR, LGBM, PK-SVM, CNN-LSTM, and TBTSC have been selected for this study since they cover complementary inductive biases and deployment profiles. ENRLR provides a regularized linear baseline with high interpretability and low training cost; LGBM represents efficient tree-based ensemble learning for structured/windowed inputs; PK-SVM represents non-linear margin-based classification; CNN-LSTM combines local temporal pattern extraction with recurrent dependency modeling; the TBTSC is characterized by attention-based modeling of long-range dependencies in multivariate time series. This heterogeneous selection is consistent with the no-free-lunch principle, according to which no single learning algorithm is expected to dominate across all problem classes [46]. It is also consistent with predictive maintenance and time series classification literature emphasizing that classical and deep models should be compared under common experimental protocols [47,48].

Although the existing literature addresses the impact of different algorithms on PdM accuracy and computational cost, only a limited number of studies have acknowledged the sensitivity of predictive performance to temporal window choices [13,14], suggesting that suboptimal horizon selection may hinder generalization or lead to misleading conclusions when comparing different models. To address this issue, this paper compares the performance of several AI approaches for PdM while varying the lengths of the observation and prediction horizons.

To better clarify the present study positioning against the existing literature, Table 1 summarizes the main research streams relevant to industrial alarm prediction, their common limitations, and the specific gaps addressed by this work. There are five main streams in which the literature can be organized: prognostics/RUL modeling, fixed-window industrial classification, ML-vs-DL predictive-maintenance benchmarks, Transformer-based time series models, and imbalanced alarm detection evaluation. These streams provide important foundations but leave a common gap, i.e., temporal framing is often fixed, underreported, or treated as a preprocessing detail. In contrast, the present study makes temporal design itself the main object of analysis by varying observation and prediction horizons across heterogeneous industrial datasets and model families.

This positioning clarifies that the present work is not primarily an additional model-ranking exercise. Instead, it uses model comparison as a controlled means to study temporal-design sensitivity. Compared with studies that evaluate a single predetermined lookback horizon or focus mainly on algorithmic architecture, this study treats the observation and prediction horizons as explicit experimental factors. Compared with prior industrial windowing studies on classification-based event prediction [13,14], the present study broadens the empirical basis by introducing a novel industrial mixing dataset, thereby enabling a clearer comparison between faster-evolving and slower-evolving process dynamics while extending the validation of the methodology to other industrial contexts.

Table 1. Research gaps addressed by this study.

Literature Stream	Reference	Focus	Limitation	Gap Addressed by This Study
PdM prognostics/RUL studies	[18–21]	Degradation trajectories and remaining useful life estimation.	Operational output differs from discrete alarm anticipation.	Focuses explicitly on binary alarm prediction within a future interval.
Fixed-window industrial classification	[28–31]	Failure or alarm classification with a chosen lookback window.	Window parameters are often fixed or weakly justified.	Treats temporal horizons as independent design variables.
ML vs. DL PdM benchmarks	[42,47]	Model-family comparison.	Temporal design is often secondary to architecture comparison.	Uses model comparison to understand sensitivity to temporal configuration.
Transformer-based time series studies	[45,47]	Attention-based temporal modeling.	Often architecture-centered and computationally costly.	Positions Transformer against simpler baselines in an industrial benchmark.
Imbalanced alarm detection	[48]	Rare-event classification metrics.	Threshold metrics often dominate; operational costs underdeveloped.	Discusses deployment implications.

3. Problem Formulation and Methodology

This section introduces the problem formulation and the methodological framework adopted in this work. Predictive maintenance is formulated as an alarm prediction task based on industrial multivariate time series, with the objective of anticipating the occurrence of abnormal events within a predefined future horizon. To this end, a window-based classification approach is employed, explicitly separating historical observations from the future prediction interval.

3.1. Problem Formulation

Let $\{x_t\}_{t=1}^T$ denote a multivariate time series acquired from an industrial system, where each observation $x_t \in \mathbb{R}^d$ represents the vector of measurements collected at time t from d sensors or process variables. In addition to sensor measurements, the system generates discrete alarm signals indicating abnormal operating conditions or faults. In this work, the predictive maintenance problem is formulated as an alarm prediction task, with the objective of anticipating the occurrence of an alarm within a predefined future time interval, based solely on historical observations.

More formally, given the time series $\{x_t\}_{t=1}^T$ and the corresponding binary alarm sequence $\{a_t\}_{t=1}^T$, where

$$a_t = \begin{cases} 1 & \text{if an alarm occurs at time } t, \\ 0 & \text{otherwise,} \end{cases}$$

the goal is to learn a predictive function

$$f : \mathbb{R}^{d \times L} \rightarrow \{0, 1\}$$

that maps a sequence of past observations of length L , corresponding to the observation horizon later referred to as the Current Window (CW), to a binary output indicating whether an alarm will occur within a future prediction horizon. Unlike regression-based formulations, which aim to estimate the RUL or the exact time to failure, this formulation focuses on event anticipation and is therefore particularly suitable for industrial environments characterized by discrete alarm mechanisms and the need for actionable early warnings.

The prediction task is thus defined as a supervised binary classification problem on time-dependent data, in which each input sample consists of a segment of multivariate

observations and the corresponding label indicates whether at least one alarm occurs within a subsequent future interval, referred to as the Future Window (FW).

3.2. Temporal Windowing and Labeling Strategy

To transform the alarm prediction problem into a supervised learning task, a window-based segmentation strategy is adopted. The continuous multivariate time series is partitioned into overlapping segments, explicitly separating past observations from future prediction intervals.

Let L denote the length of the CW, representing the observation horizon, and H the length of the Future Window (FW), representing the prediction horizon. For each time index t , an input sample is constructed from the past L observations:

$$X_t = \{x_{t-L+1}, x_{t-L+2}, \dots, x_t\} \in \mathbb{R}^{d \times L}.$$

The corresponding target label y_t is determined based on the occurrence of alarms within the subsequent future interval of length H :

$$y_t = \begin{cases} 1 & \text{if } \exists \tau \in [t+1, t+H] \text{ such that } a_\tau = 1, \\ 0 & \text{otherwise.} \end{cases}$$

In this formulation, a positive label indicates that at least one alarm occurs within the future window, whereas a negative label corresponds to normal operation throughout the entire prediction horizon.

The dataset used for model training is obtained by sliding the pair (X_t, y_t) along the time axis with a fixed stride, thereby generating a set of input–output pairs suitable for supervised classification. Overlapping windows allow the model to capture fine-grained temporal evolution of the system while preserving continuity in the underlying process dynamics.

The explicit separation between observation and prediction intervals ensures that no future information leaks into the input representation, thereby maintaining a realistic early-warning scenario.

Unlike conventional approaches in which temporal window lengths are fixed a priori, this work treats L (CW) and H (FW) as independent design variables. By systematically varying both parameters, it becomes possible to investigate how the amount of historical information and the anticipation horizon jointly influence predictive performance and operational lead time.

This formulation enables a controlled analysis of the trade-off between:

- Longer observation windows, which may capture slower degradation trends and complex temporal dependencies;
- Shorter observation windows, which favor responsiveness and computational efficiency;
- Longer prediction horizons, which increase actionable lead time but may introduce greater uncertainty;
- Shorter prediction horizons, which typically yield more reliable predictions at the expense of reduced anticipation capability.

The impact of these temporal design choices on different machine learning and deep learning models is evaluated in the experimental section.

3.3. Learning Framework and Experimental Pipeline

The alarm prediction problem defined in the previous subsections is addressed through a unified learning framework designed to enable a controlled evaluation of temporal window parameters across heterogeneous model families.

After generating the set of input–output pairs (X_t, y_t) , the resulting dataset is used to train supervised predictive models. Each sample consists of a multivariate observation window $X_t \in \mathbb{R}^{d \times L}$ and the corresponding binary label y_t , indicating whether an alarm occurs within the future window of length H .

The learning pipeline follows a standardized structure composed of the following stages:

1. Window extraction and labeling: construction of input–output pairs using predefined length values of CW and FW.
2. Dataset partitioning: separation of data into training, validation, and test subsets while preserving temporal ordering to ensure a realistic evaluation scenario.
3. Model training: supervised learning models are trained using the training subset, while the validation subset is used for hyperparameter tuning.
4. Performance evaluation: predictive performance is assessed on the test subset using appropriate classification metrics.

To ensure comparability, all models are evaluated under the same data partitioning strategy and identical window configurations. This allows the analysis to isolate the effect of temporal parameters from model-specific characteristics.

The experimental study includes both machine learning and deep learning approaches, enabling the assessment of how different model families respond to variations in observation and prediction horizons. Classical machine learning models operate on vectorized representations of the input windows, whereas deep learning models directly extract the features from the temporal structure of the multivariate sequences.

Crucially, the parameters L and H are treated as independent experimental variables rather than fixed preprocessing choices. By systematically varying these parameters over a predefined range of values, it becomes possible to investigate three fundamental aspects: the sensitivity of predictive performance to the amount of historical information available to the model, the effect of anticipation length on classification performance, and the interaction between temporal horizon design and model architecture.

This experimental framework enables a structured investigation of the trade-off between early warning capability and predictive accuracy, providing insights into the temporal configuration that best balances operational lead time and model robustness in industrial alarm prediction scenarios. Figure 1 provides an overview of the proposed methodology.

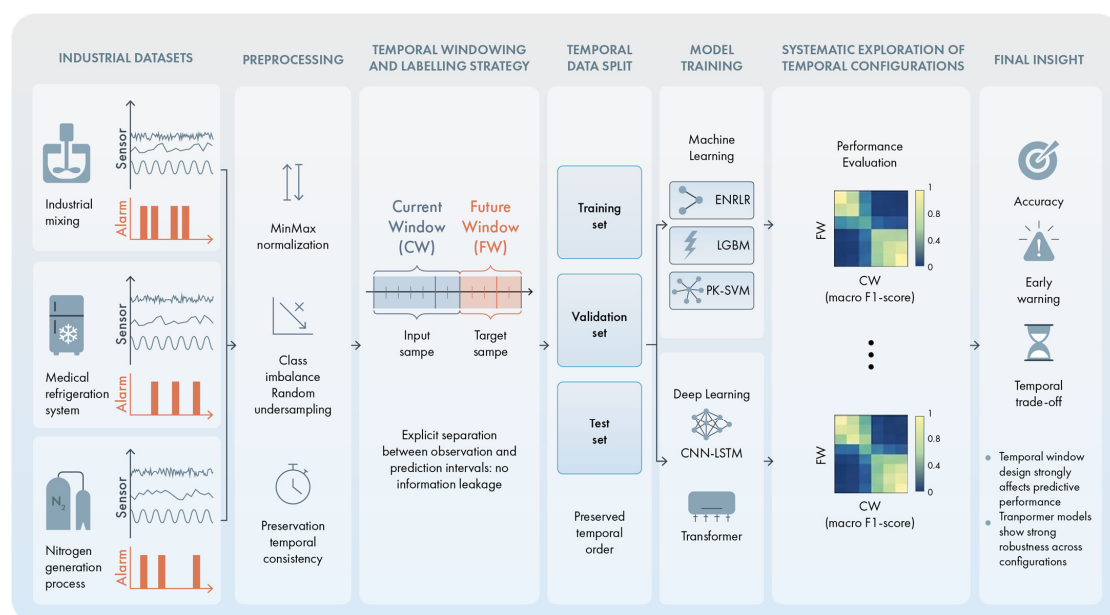


Figure 1. Framework of the employed methodology.

The specific model configurations, hyperparameters, and evaluation metrics are detailed in the following section.

4. Experimental Setup

The methodology was applied to three multivariate time series datasets collected from distinct industrial contexts. The first dataset was provided by an industrial company located in southern Italy. It contains data from an industrial mixing process, in which sensor measurements describe the evolution of physical and operational variables during production cycles. Table A1 in Appendix A reports the parameters included in the dataset together with their units of measurement or notation. The data cannot be disclosed because it is proprietary to the company.

The second dataset concerns a refrigeration system used in a medical context, where the monitored variables capture thermodynamic and other operational conditions of the machine. Among the 58 monitored parameters, only 13 have been identified as relevant [14] and selected for use in the present study. These variables are listed in Table A2 in Appendix A, together with their units of measurement or notation and the minimum and maximum values over the two-month monitoring period.

The third dataset concerns a nitrogen generation process, characterized by slower dynamics and longer-term dependencies among variables. Five parameters have been selected from this dataset [14], and are reported in Table A3 in Appendix A.

In all cases, the datasets consist of synchronized sensor measurements acquired over time, where each observation corresponds to a vector of process variables. In addition to sensor data, each dataset includes a binary alarm signal indicating the occurrence of abnormal events. Together, these datasets provide a heterogeneous benchmark for evaluating the robustness of predictive models across different industrial scenarios.

Prior to model training, a preprocessing pipeline is applied to ensure consistency and reliability of the input data. All features are normalized using Min–Max scaling [49], which preserves the relative structure of the signals while ensuring numerical stability during model training. The normalization parameters are computed exclusively on the training set and subsequently applied to validation and test sets, thereby preventing any form of data leakage. Given the inherent imbalance between normal operating conditions and alarm events, appropriate strategies are adopted to mitigate the effect of class imbalance during training, including random undersampling. Random undersampling of the majority class is used as a pragmatic way to reduce class dominance and computational burden across many temporal configurations. This choice preserves the original sensor values and keeps the benchmark feasible.

The predictive performance is evaluated using a set of supervised learning models belonging to both machine learning and deep learning families. The selected machine learning models include Elastic-Net Regularized Logistic Regression (ENRLR) [35], Light Gradient Boosting Machine (LGBM) [36], and Polynomial Kernel Support Vector Machine (PK-SVM) [50], chosen for their effectiveness on structured tabular data and in industrial applications. In addition, deep learning architectures such as Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) [41] and Transformer-based models [45] are considered, as they can capture temporal dependencies and complex patterns in multivariate time series data.

ENRLR was implemented using an Elastic-Net penalized logistic regression model. This formulation combines L1 and L2 regularization, enabling both coefficient shrinkage and feature selection. The model was optimized using the SAGA solver [51]. The optimization procedure used a maximum of 20,000 iterations and a convergence tolerance of 10^{-3} .

Hyperparameter tuning was performed over the regularization strength C [52] and the Elastic-Net mixing parameter $l1_ratio$.

LGBM was optimized using the active hyperparameter grid, including number of estimators, learning rate, number of leaves, and maximum depth of each decision tree. These parameters control the number of boosting iterations, the contribution of each tree, and the complexity of the individual decision trees. Early stopping was applied using validation log-loss to reduce overfitting.

PK-SVM was implemented to capture nonlinear decision boundaries through polynomial feature mapping. Hyperparameter optimization was conducted using a 3-fold stratified grid search over C , degree (d), gamma (γ), and $coef0$, corresponding to the regularization strength and the main polynomial-kernel parameters [53].

The CNN-LSTM architecture was designed to capture both local patterns and sequential dependencies in the input representation. A one-dimensional convolutional layer was followed by max-pooling to extract compact local features, which were then processed by an LSTM layer. Dropout and a dense ReLU layer with L2 regularization were included to improve generalization, followed by a sigmoid output layer for binary classification. Bayesian optimization was used to tune the number of convolutional filters, kernel size, pooling size, LSTM units, dropout rate, dense units, L2 regularization coefficient, and learning rate.

The Transformer-based model was implemented as a single encoder-style block. Input features were first projected into an embedding space and combined with sinusoidal positional encodings to preserve sequential information. The encoder block consisted of multi-head self-attention, a point-wise feed-forward layer, residual connections, and layer normalization. Global average pooling and dropout with a rate of 0.1 were applied before the final sigmoid output layer. Optuna [54] was used to tune the embedding dimension, valid attention-head configurations, feed-forward dimension, and learning rate.

Model training was performed using a temporal data splitting strategy, in which the datasets are divided into training, validation, and test subsets while preserving chronological order. This approach reflects real-world deployment conditions and avoids information leakage from future observations. Hyperparameter selection is conducted using the validation set, while final performance is assessed on the test set.

The experimental models were implemented in Python 3.12 using established machine learning and deep learning libraries. The conventional machine learning models relied primarily on scikit-learn [55], with ENRLR implemented through LogisticRegression and PK-SVM through SVC. The LGBM model was implemented using LightGBM's LGBM-Classifer, which offers a scikit-learn-compatible estimator interface. For the deep learning approaches, the CNN-LSTM architecture was developed using TensorFlow/Keras and optimized with KerasTuner, whereas the Transformer-based architecture was implemented in PyTorch 2.7 and tuned using Optuna.

Model performance is primarily evaluated using the F1-macro score [56], which provides a balanced measure of precision and recall across the two classes. This metric is particularly suitable for imbalanced datasets, in which alarm events are substantially less frequent than normal operating conditions. The F1-macro score ensures that both false positives and false negatives are appropriately penalized. This choice reflects the operational requirements of industrial alarm prediction, where missed alarms and excessive false alarms can both have critical consequences.

A key aspect of the experimental design is the systematic exploration of temporal window parameters. Specifically, multiple CW-FW configurations are considered, where CW represents the observation horizon and FW represents the prediction horizon. The evaluated configurations are reported in Table 2.

Table 2. CW–FW configurations used for each dataset.

Dataset	Current Window (CW) [Minutes]	Future Window (FW) [Minutes]
Mixing Process	10, 15, 20, 25, 30, 45, 60	30, 60, 90
Refrigeration System	10, 15, 20, 25, 30, 45, 60	30, 60, 90, 120
Nitrogen Generation Process	10, 15, 20, 25, 30, 45, 60	30, 60, 90, 120, 180, 300

These parameters are varied independently across a predefined range of values, enabling the analysis of their individual and joint effects on predictive performance.

5. Results

The experimental results provide a comprehensive evaluation of the proposed methodology across multiple datasets, models, and temporal configurations. In particular, the analysis examines the effects of several CW and FW configurations on predictive performance, as well as on the comparative behavior of different machine and deep learning models.

Table 3 reports the average training time, in minutes, across the tested CW–FW configurations for each dataset and model. Training was performed on an NVIDIA GeForce GTX 1070 Ti for the deep learning models and an AMD Ryzen 5 5600X for the classical ML models. The results show that computational burden increases with model and dataset complexity. In particular, the Transformer is the most expensive model in terms of average training time, whereas ENRLR and LGBM provide substantially faster alternatives. Consequently, when two configurations achieve comparable F1–macro scores, the lower-cost model should be preferred for industrial deployment. From an implementation perspective, ENRLR and LGBM are expected to have the smallest deployment footprint; PK-SVM may become more costly at inference because prediction depends on support vectors; CNN-LSTM requires neural network weights and sequence activations during training; the Transformer-based model is characterized by the largest expected training time footprint, since self-attention introduces sequence-length-dependent memory and compute costs [57]. A future deployment benchmark should log wall-clock training time, peak CPU/GPU memory, inference latency, throughput, and energy consumption under identical hardware, batch size, and software constraints.

Table 3. Average training time expressed in minutes.

Dataset	ENRLR	LGBM	PK-SVM	CNN-LSTM	Transformer
Mixing Process	2.27	2.42	3.43	10.21	23.02
Refrigeration System	0.83	0.86	2.20	5.40	8.20
Nitrogen Generation Process	0.30	0.46	1.47	2.99	6.68

It is worth noting that the training time in the mixing process dataset is higher than the other two considered datasets because of its higher complexity.

The results are presented through heatmaps. Each heatmap represents the F1–macro score as a function of CW and FW, providing a visual overview of how predictive performance varies across different temporal configurations. The use of a common color scale across all heatmaps enables direct comparison of model behavior and performance trends.

5.1. Mixing Process

Figure 2 shows the results obtained for the mixing process dataset. The performance heatmaps highlight a clear dependency of predictive performance on both CW and FW, as well as significant differences across models. A consistent trend observed across all models is the degradation in performance as the prediction horizon increases. In particular, moving

from FW = 30 min to FW = 90 min generally leads to lower F1-macro scores, confirming that anticipating alarms further into the future introduces greater uncertainty and reduces classification reliability. This effect is evident in both machine and deep learning models and is particularly pronounced for CNN-LSTM and ENRLR, where performance drops are more pronounced at larger FW values.

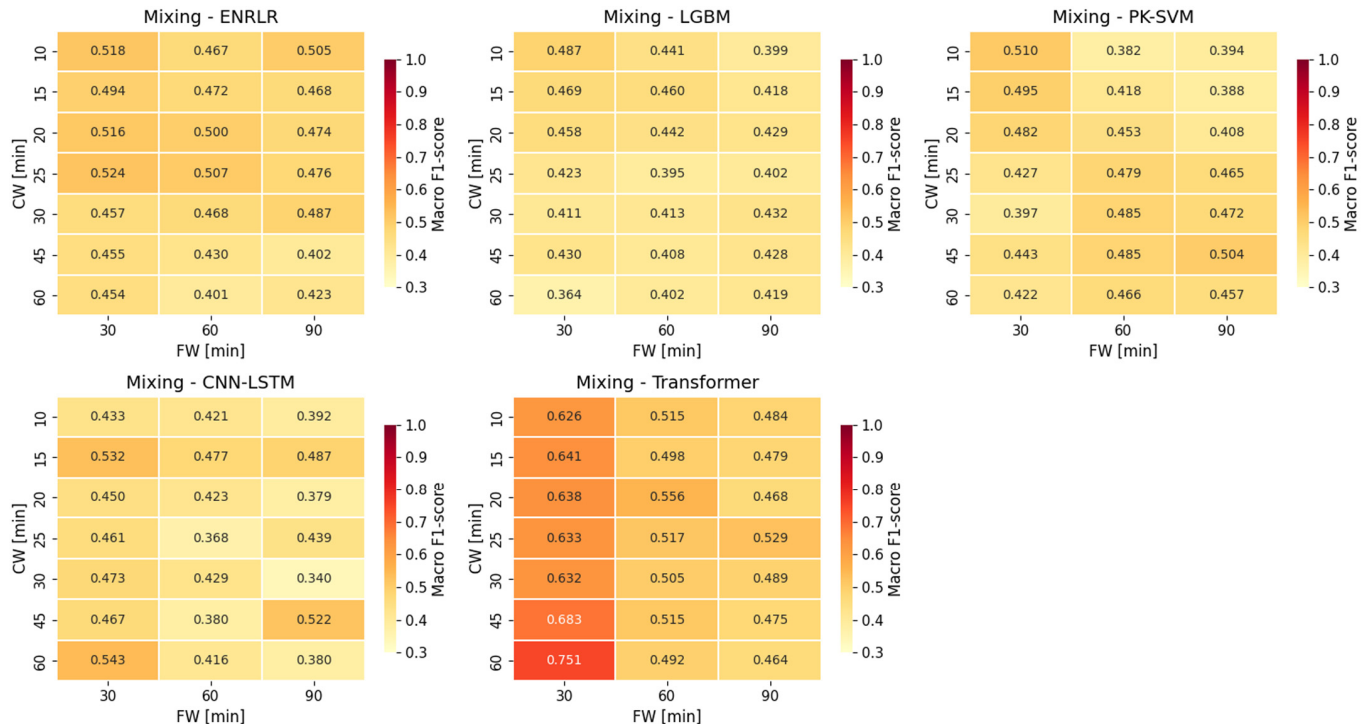


Figure 2. Heatmaps of the F1-macro score for the mixing process.

The effect of the observation window (CW) appears to be more nuanced and model-dependent. For classical machine learning models such as ENRLR and LGBM, performance tends to remain relatively stable across different CW values, with slight improvements observed for intermediate window lengths (e.g., CW between 20 and 30 min). In contrast, PK-SVM shows a noticeable sensitivity to CW, achieving better performance with larger observation windows, especially when combined with longer prediction horizons.

Deep learning models exhibit a different behavior. The CNN-LSTM architecture shows moderate variability with respect to CW, with no clearly dominant configuration, suggesting limited ability to fully exploit longer temporal contexts in this dataset. On the other hand, the Transformer-based model consistently outperforms all other approaches across almost all CW–FW configurations. In particular, it achieves the highest performance values for shorter prediction horizons (FW = 30 min), with performance improving further as the observation window increases, reaching peak values for larger CW configurations.

Overall, the results indicate that for the mixing process, shorter prediction horizons combined with sufficiently large observation windows lead to the best predictive performance.

Table 4 highlights the best-performing model and the corresponding F1-macro score for each CW–FW configuration.

The superior performance and stability of the Transformer model suggest that attention-based architectures are particularly effective in capturing the temporal dependencies of this process. However, for some longer prediction horizons, especially at FW = 90 min, classical or hybrid models such as ENRLR and CNN-LSTM occasionally achieve the best score. This suggests that, when the prediction task becomes more uncer-

tain, the advantage of the Transformer-based model is reduced, and simpler models may remain competitive for specific temporal configurations.

Table 4. Best-performing models (F1–macro score) for every CW–FW configuration in the context of mixing.

CW/FW	30 min	60 min	90 min
10 min	Transformer (0.626)	Transformer (0.515)	ENRLR (0.505)
15 min	Transformer (0.641)	Transformer (0.498)	CNN-LSTM (0.487)
20 min	Transformer (0.638)	Transformer (0.556)	ENRLR (0.474)
25 min	Transformer (0.633)	Transformer (0.517)	Transformer (0.529)
30 min	Transformer (0.632)	Transformer (0.505)	Transformer (0.489)
45 min	Transformer (0.683)	Transformer (0.515)	CNN-LSTM (0.522)
60 min	Transformer (0.751)	Transformer (0.492)	Transformer (0.464)

5.2. Refrigeration System

For the refrigeration system dataset, whose results are shown in Figure 3, the performance heatmaps confirm a strong dependence of predictive performance on both the CW and FW, while also revealing a generally better performance compared to the mixing process.

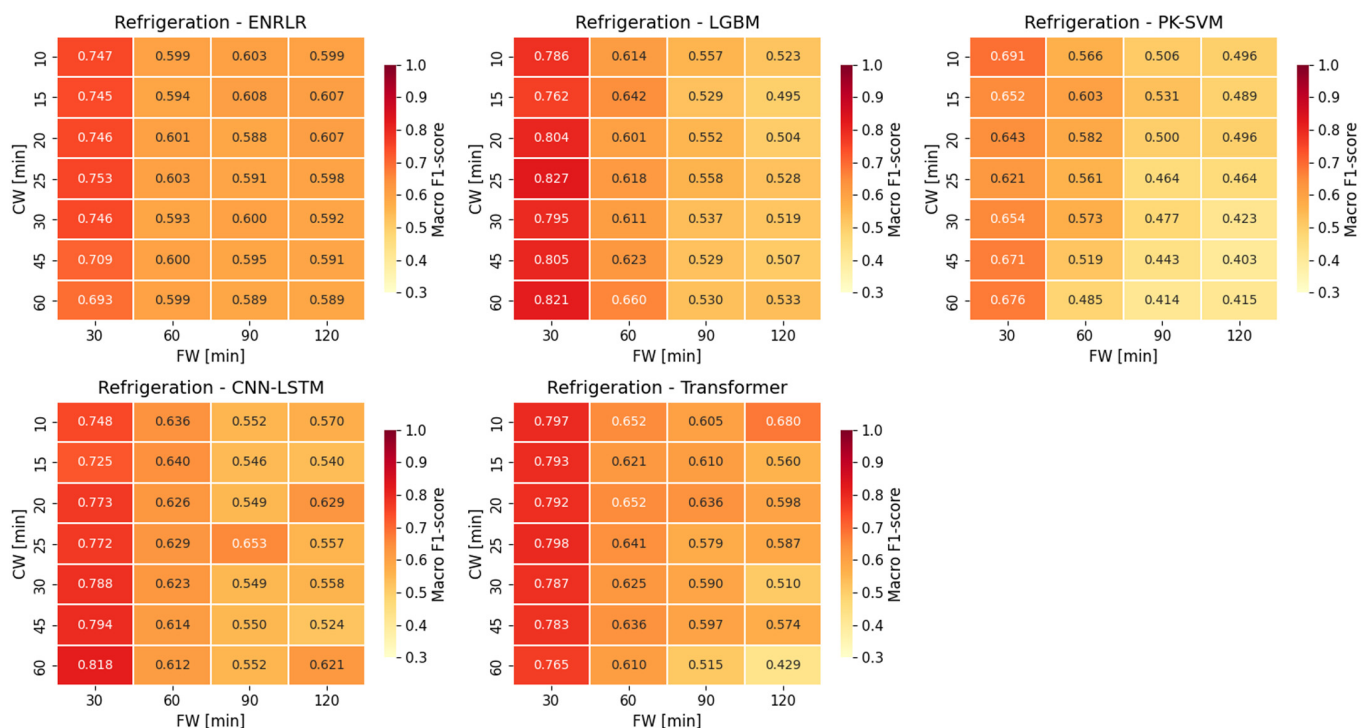


Figure 3. Heatmaps of the F1–macro score for the refrigeration system.

Similarly to the previous case, a consistent decrease in performance is observed as the prediction horizon increases. Across all models, moving from FW = 30 min to FW = 120 min results in a progressive reduction in F1–macro score values, highlighting the increasing difficulty of anticipating alarms further in advance. However, compared with the mixing

process dataset, this degradation is less abrupt, suggesting that the refrigeration system exhibits more stable and predictable dynamics over longer time horizons.

The influence of the observation window (CW) is more evident in this dataset. For machine learning models, particularly LGBM, performance improves as CW increases, with the highest scores obtained for larger observation windows. This indicates that longer temporal contexts are beneficial for capturing the underlying system's behavior. ENRLR shows a more stable performance across CW values, while PK-SVM exhibits moderate sensitivity, with slightly better results for intermediate window lengths.

Deep learning models further emphasize the importance of longer observation horizons. The CNN-LSTM architecture shows a clear performance improvement as CW increases, reaching its best results for the largest window sizes. The Transformer-based model again demonstrates strong and stable performance across a wide range of configurations, maintaining high F1-macro scores even for longer prediction horizons. In particular, it exhibits greater robustness to changes in FW than the other models, indicating a superior ability to capture longer-term temporal dependencies.

Overall, the results suggest that, for the refrigeration dataset, larger observation windows combined with shorter prediction horizons lead to the best predictive performance, although acceptable results can still be obtained for longer FW values. Compared with the mixing process dataset, the stronger impact of CW highlights the relevance of longer temporal dependencies in this system.

Table 5 highlights the best-performing model and the corresponding F1-macro score for each CW-FW configuration.

Table 5. Best-performing models (F1-macro score) for every CW-FW configuration in the context of the refrigeration system.

CW/FW	30 min	60 min	90 min	120 min
10 min	Transformer (0.797)	Transformer (0.652)	Transformer (0.605)	Transformer (0.680)
15 min	Transformer (0.793)	LGBM (0.642)	Transformer (0.610)	ENRLR (0.607)
20 min	LGBM (0.804)	Transformer (0.652)	Transformer (0.636)	CNN-LSTM (0.629)
25 min	LGBM (0.827)	Transformer (0.641)	CNN-LSTM (0.653)	ENRLR (0.598)
30 min	LGBM (0.795)	Transformer (0.625)	ENRLR (0.600)	ENRLR (0.592)
45 min	LGBM (0.805)	Transformer (0.636)	Transformer (0.597)	ENRLR (0.591)
60 min	LGBM (0.821)	LGBM (0.660)	ENRLR (0.589)	CNN-LSTM (0.621)

While the Transformer-based model frequently achieves the highest F1-macro score, especially for shorter and intermediate prediction horizons, LGBM becomes particularly competitive when larger observation windows are considered, especially at FW = 30 min. This confirms the behavior observed in the heatmaps, where the refrigeration system benefits from longer observation windows and where tree-based ensemble learning can effectively exploit structured temporal features extracted from the windows. ENRLR and CNN-LSTM also emerge as best models in some configurations involving longer FW values, indicating that no single model dominates uniformly across the entire temporal space.

5.3. Nitrogen Generation Process

For the nitrogen generation dataset, the performance heatmaps (Figure 4) reveal a markedly different behavior compared to the previous cases, with substantially higher predictive performance across all models and a stronger dependence on CW.

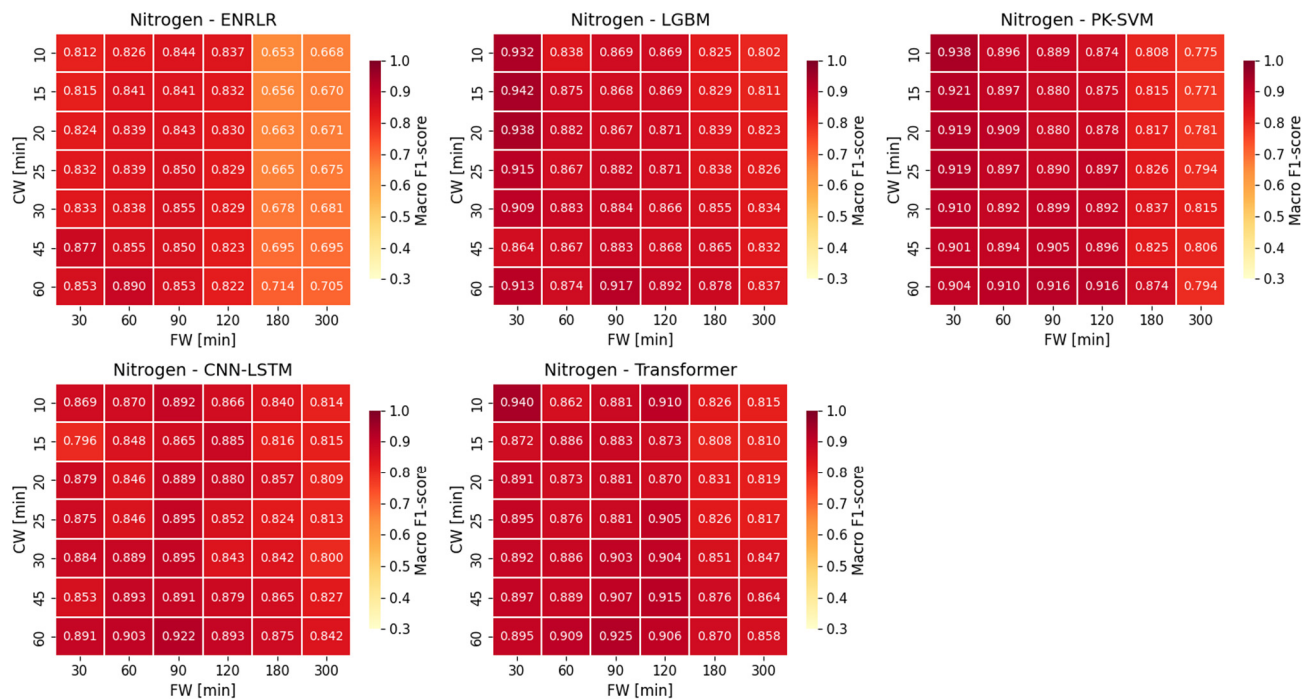


Figure 4. Heatmaps of the F1–macro score for the nitrogen generation process.

As observed in the other datasets, predictive performance generally decreases as the prediction horizon increases. However, in this case, the degradation with respect to FW is less pronounced, and relatively high F1–macro score values are maintained even for long prediction horizons (up to FW = 300 min). This suggests that the nitrogen generation process exhibits more regular and slowly evolving dynamics, allowing models to predict alarms reliably over longer time intervals. The multivariate time series exhibits a clear distinction between normal and abnormal dynamics in the monitored parameters.

The effect of the observation window is particularly evident in this dataset. Across all models, increasing CW leads to noticeable improvements in predictive performance, with the best results consistently obtained for larger observation windows. This trend indicates that long temporal contexts are essential for capturing the underlying behavior of the system, which is characterized by gradual changes and long-term dependencies.

Among machine learning models, LGBM and PK-SVM achieve strong performance across a wide range of configurations, with PK-SVM showing particularly competitive results for intermediate and large CW values. ENRLR also performs well, although with slightly lower peak values compared to the other approaches. Deep learning models further support this trend: CNN-LSTM exhibits high performance for larger CW values, while the Transformer-based model demonstrates both high accuracy and strong robustness across different CW–FW configurations. In particular, the Transformer-based architecture achieves some of the highest overall scores and maintains stable performance even for extended prediction horizons.

Overall, the results indicate that, for the nitrogen generation process, large observation windows combined with both short and long prediction horizons lead to consistently high predictive performance. Compared with the mixing and refrigeration datasets, the

reduced sensitivity to FW and the stronger dependence on CW highlight the importance of long-term temporal information in this system.

Table 6 highlights the best-performing model and the corresponding F1-macro score for each CW–FW configuration.

Table 6. Best-performing models (F1-macro score) for every CW–FW configuration in the context of nitrogen generation.

CW/FW	30 min	60 min	90 min	120 min	180 min	300 min
10 min	Transformer (0.940)	PK-SVM (0.896)	CNN-LSTM (0.892)	Transformer (0.910)	CNN-LSTM (0.840)	Transformer (0.815)
15 min	LGBM (0.942)	PK-SVM (0.897)	Transformer (0.883)	CNN-LSTM (0.885)	LGBM (0.829)	CNN-LSTM (0.815)
20 min	LGBM (0.938)	PK-SVM (0.909)	CNN-LSTM (0.889)	CNN-LSTM (0.880)	CNN-LSTM (0.857)	LGBM (0.823)
25 min	PK-SVM (0.919)	PK-SVM (0.897)	CNN-LSTM (0.895)	Transformer (0.905)	LGBM (0.838)	LGBM (0.826)
30 min	PK-SVM (0.910)	PK-SVM (0.892)	Transformer (0.903)	Transformer (0.904)	LGBM (0.855)	Transformer (0.847)
45 min	PK-SVM (0.901)	PK-SVM (0.894)	Transformer (0.907)	Transformer (0.915)	Transformer (0.876)	Transformer (0.864)
60 min	LGBM (0.913)	PK-SVM (0.910)	Transformer (0.925)	PK-SVM (0.916)	LGBM (0.878)	Transformer (0.858)

The best-performing model varies more frequently among LGBM, PK-SVM, CNN-LSTM, and the Transformer-based model. This result suggests that this process contains strong and relatively stable predictive patterns that can be captured by both classical and deep learning models. The frequent presence of PK-SVM and LGBM confirms the effectiveness of nonlinear machine learning approaches in this dataset, while the Transformer-based model remains highly competitive across several long-horizon configurations.

6. Discussion

This section interprets the experimental results from theoretical and managerial perspectives. Theoretically, the findings indicate that temporal windowing is not a neutral preprocessing step. The observation window controls how much historical process information is available to the classifier, whereas the future window controls the time interval over which the model must anticipate an alarm. Their interaction affects the effective signal-to-noise ratio of the classification task. From a managerial perspective, the prediction horizon should be aligned with the minimum lead time needed for inspection, intervention, or safe shutdown, while the observation horizon should be selected according to process dynamics and computational constraints.

6.1. Theoretical Implications

The results obtained across the three datasets provide consistent and complementary insights into the role of temporal window design in PdM tasks. In particular, the joint analysis of CW and FW highlights that predictive performance is not determined by these parameters independently, but by their interaction, confirming the importance of treating temporal windowing as a core component of the modeling process rather than a simple preprocessing step.

The first key finding concerns the impact of the prediction horizon. Across all datasets and models, an increase in FW generally leads to a reduction in predictive performance, which is consistent with existing literature on time-series prediction and failure forecasting [24,26]. Longer prediction horizons introduce higher uncertainty and reduce the

separability between normal and pre-alarm conditions, making classification more challenging. However, the extent of this degradation strongly depends on the underlying process dynamics. While the mixing process shows a rapid performance drop for increasing FW, the refrigeration system exhibits a more gradual degradation, and the nitrogen generation process maintains high performance even for long prediction horizons. This behavior aligns with studies highlighting that systems characterized by slower and more regular dynamics allow more reliable long-term predictions [58].

This result is consistent with previous work on industrial failure prediction showing that prediction-window length can have a strong effect on model performance [13,14]. The present results extend this evidence by showing that the magnitude of the performance reduction is process-dependent. In the mixing dataset, alarms appear more difficult to anticipate at longer horizons, which is consistent with faster or more variable dynamics. In contrast, the nitrogen generation process maintains higher F1-macro scores at extended FW values, suggesting that its alarm precursors are more regular or evolve over longer characteristic time scales. Therefore, the practical meaning of a “long” prediction horizon is not universal; it depends on the dynamics of the physical process being monitored.

A second important observation relates to the role of the observation window. The results clearly show that the benefit of increasing CW is highly dependent on the temporal characteristics of the system. For systems with fast dynamics, such as the mixing dataset, increasing CW provides limited improvements, suggesting that relevant information is concentrated in shorter temporal contexts. In contrast, for systems with slower dynamics, such as the refrigeration system and especially the nitrogen generation process, larger observation windows substantially improve predictive performance. This finding is consistent with previous works emphasizing the importance of long historical contexts for capturing degradation trends and temporal dependencies [26,58], while also confirming the trade-off between performance gains and computational cost discussed in [34].

The role of CW should therefore be interpreted as a process-dynamics question rather than as a monotonic modeling rule. Longer CW values can improve performance when they capture slow degradation patterns or delayed dependencies, but they can also add redundant or stale information in faster processes. This explains why the nitrogen generation process benefits more consistently from larger CW values, whereas the mixing process shows more limited gains. This interpretation strengthens the main theoretical contribution of the paper: temporal configuration should be selected as part of the predictive-maintenance design, not inherited as a fixed preprocessing convention.

The results suggest that temporal window effects are process-dependent rather than monotonic. Increasing the prediction horizon generally reduces F1-macro because the model is required to anticipate events farther away from the available observations. As FW increases, uncertainty about intervening operating conditions grows, and the observed sensor pattern may become less directly related to the eventual alarm. This agrees with prior industrial failure prediction work showing that prediction window size can strongly affect model performance [14]. The effect of CW is more nuanced. Longer observation windows provide more historical context and can help when alarm precursors evolve gradually or when the process has long characteristic time constants. This explains why slowly evolving processes such as nitrogen generation benefit from larger CW values. Conversely, in faster or cyclical systems, additional history can add stale information, increase dimensionality, and reduce the benefit of flattened representations. Thus, CW should be aligned with process dynamics rather than treated as a fixed preprocessing parameter.

From a model perspective, the results highlight clear differences between machine and deep learning approaches. Classical machine learning models, such as ENRLR and LGBM, generally provide stable and competitive performance, confirming their effec-

tiveness in structured industrial data scenarios, as reported in [35,36]. However, their sensitivity to temporal configurations is often limited, suggesting a reduced ability to fully exploit complex temporal dependencies. In contrast, deep learning models, particularly the Transformer-based architecture, demonstrate superior performance and robustness across different CW–FW configurations. This behavior is consistent with recent studies showing the effectiveness of attention-based models in capturing long-range dependencies in multivariate time series [45]. At the same time, CNN-LSTM models exhibit more variable performance, reflecting the challenges associated with modeling heterogeneous and noisy industrial data using hybrid architectures [44].

The model-specific patterns reflect model inductive biases. ENRLR and LGBM remain competitive when alarm patterns are regular or sufficiently represented by flattened windows. CNN-LSTM can exploit local motifs and recurrent dependencies when those patterns are present. The Transformer-based model can remain robust when cues are distributed across longer histories because self-attention can assign different weights to distant time steps. However, the flexibility of deep models must be weighed against their higher computational cost.

The results support the idea that there is no universally optimal temporal configuration, but rather that the choice of CW and FW must be adapted to the specific characteristics of the industrial system. This observation addresses a key limitation of many existing studies, where temporal windows are fixed a priori and not systematically analyzed [28–31]. By explicitly varying both parameters and evaluating their joint impact, this work provided a more comprehensive understanding of the trade-offs involved in predictive maintenance design.

6.2. Managerial Implications

From an operational and managerial perspective, the findings have important implications. The selection of temporal horizons directly affects not only predictive accuracy but also the balance between early warning capability and decision reliability. Shorter prediction horizons may lead to more accurate predictions, but provide less actionable time, while longer horizons increase anticipation but at the cost of higher uncertainty. Similarly, longer observation windows improve performance in slow-dynamics systems but may increase computational requirements and latency. Hence, FW should first be anchored to the minimum actionable lead time required for inspection, intervention, or safe shutdown. CW should then be tuned around the characteristic time scale of the process. For fast systems, shorter observation windows may be sufficient and cheaper; for slow-dynamic systems, longer observation windows may be necessary to capture gradual precursors. The results also suggest a deployment rule, i.e., when F1–macro is similar, prefer the simpler and faster model; reserve deep architectures for cases where long-range temporal structure provides a clear benefit. In safety-critical settings, the model should be used as a decision support component only after threshold calibration, cost-sensitive validation, and human-in-the-loop review. Trade-offs must therefore be carefully considered when deploying predictive maintenance solutions in real industrial environments.

In deployment, the best temporal configuration is not necessarily the one with the highest isolated F1–macro score. A configuration must also provide enough lead time for action, maintain an acceptable false-alarm burden, and fit the available computational resources. For imbalanced alarm prediction, future deployments should therefore complement F1–macro score with threshold-sensitive and cost-sensitive analyses, such as precision-recall curves, average precision, alert burden, missed-alarm cost, and process-specific cost matrices. This is particularly important because industrial false alarms and missed alarms usually have asymmetric operational consequences.

Finally, computational times should also influence deployment. Table 3 shows that the Transformer-based model is the most computationally expensive option, whereas ENRLR and LGBM provide faster alternatives. Thus, the best industrial configuration is not necessarily the one with the highest absolute F1-macro, but the one that provides sufficient lead time and predictive reliability under available compute, operational risk, and human-review constraints.

7. Conclusions

This paper investigated the impact of temporal window design on predictive maintenance performance, formulating alarm prediction as a window-based classification problem on industrial multivariate time series. In particular, the study focused on the joint effect of the observation horizon (CW) and the prediction horizon (FW), systematically analyzing their influence across different datasets and predictive models.

The results demonstrated that predictive performance is strongly dependent on the interaction between observation and prediction horizons, rather than on their individual values. While shorter prediction horizons generally lead to higher predictive accuracy, their effectiveness depends on the temporal dynamics of the underlying process. Similarly, the benefit of increasing the observation window varies across systems, being limited in fast-dynamics processes and substantially more relevant in slow-dynamics scenarios.

The comparative evaluation of machine and deep learning models further highlighted that attention-based architectures, such as Transformer-based models, provide superior performance and robustness across a wide range of temporal configurations, while classical models remain competitive in less complex settings with lower computational requirements.

Overall, the findings confirm that temporal window selection is a critical design factor in predictive maintenance based on classification, and should be explicitly considered during model development and deployment. By providing a systematic analysis of CW-FW configurations, this work contributed to a better understanding of the trade-offs between predictive accuracy, anticipation capability, and computational complexity in industrial alarm prediction.

This study has several limitations. One limitation is the absence of formal statistical validation of performance differences. The experiments were designed as a controlled temporal holdout comparison across datasets, models, and CW-FW configurations, but repeated runs, blocked bootstrap samples, rolling-origin folds, or complete per-window score vectors were not retained. Consequently, it is not possible to compute reliable confidence intervals or significance tests from the aggregate F1-macro tables alone. This limitation is important since overlapping windows are not independent observations, and standard random cross-validation may violate temporal ordering. Another limitation of this study is the use of F1-macro score alone to assess the results. F1-macro does not provide a complete threshold-dependent view of classifier behavior. Precision-Recall analysis and average precision would be particularly useful complementary metrics in imbalanced binary classification, while ROC-AUC (Receiver Operating Characteristic Area Under the Curve) would provide a broader ranking-oriented view. These metrics have not been considered in the present study, since the retained experiment archive does not contain the per-window probability or decision score vectors required to reconstruct curves for every dataset, model, and CW-FW configuration. Similarly, cost-sensitive evaluation would require a process-specific cost matrix assigning operational or economic weights to false alarms and missed failures. This information was missing from the analyzed datasets.

Future work may extend this approach by incorporating new data to evaluate additional metrics such as ROC-AUC, expected cost, alert burden, and missed-failure cost. In addition, adaptive or data-driven strategies for dynamic window selection may be

considered, as well as exploring the integration of temporal optimization techniques into automated model selection frameworks. Further investigations into larger and more heterogeneous datasets may help validate the generalizability of the proposed methodology. Finally, given that random undersampling may not always be the optimal choice to mitigate class imbalance, alternative approaches, such as class-weighted fitting, focal loss, synthetic oversampling, threshold calibration, and cost-sensitive optimization could be compared.

Author Contributions: Conceptualization, T.U., M.F. and R.E.L.; methodology, M.C. and T.U.; software, T.U. and R.E.L.; formal analysis, M.C., A.N. and D.S.; investigation, T.U., I.S., R.E.L. and F.S.; resources, R.M., I.S., S.M., D.S. and L.M.; data curation, I.S., F.S., T.U. and R.E.L.; writing—original draft preparation, M.C., M.F. and T.U.; writing—review and editing, S.M., R.M., R.E.L., A.N. and L.M.; visualization, M.C. and T.U.; supervision, R.M. and S.M.; project administration, D.S., A.N., L.M., M.F. and S.M.; funding acquisition, D.S., M.F. and I.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Enterprises and Made in Italy (MIMIT), under IMPROVE (decIision support systeM for Production management based on pRedcitiOn of high impact VariableS) project, GA F/350401/01-05/X60 Decree No. 2116 of 3 December 2024.

Data Availability Statement: The industrial mixing dataset is not publicly available due to company confidentiality restrictions. The refrigeration and nitrogen generation datasets analyzed in this study are available from the source cited in [14]. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Riccardo Emanuele Landi, Luca Mozzillo, Antonio Negro, Domenico Soriano, were employed by the company Mare Engineering Group S.p.A.; author Federico Saponaro, Ivo Surano were employed by the company Gelesis Srl. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

Table A1. Monitored variables for the industrial mixing process.

Parameter	Description	Unit Symbol/Notation
CM_Agitator_001.AmpsPV	Agitator Current	A
CM_Agitator_001.PV	Agitator Revolutions per Minute	RPM
CM_Agitator_001.StateRunning	Agitator Status	N/A [Binary]
CM_M166001_001.AmpsPV	Vacuum Pump Current	A
CM_M166001_001.StateRunning	Vacuum Pump Status	N/A [Binary]
CM_PIC161001_001.PV	Tank Pressure	bar
CM_TIC162001_001.PV	Tank Temperature	°C
CM_VesselWeight_001.PV	Tank Mass	kg
UN_Mixer_Ekato_001.Run	Mixing Status	N/A [Binary]
UN_Mixer_Ekato_001.AlarmActive	Alarm Status	N/A [Binary]

Table A2. Monitored variables for the refrigeration system.

Parameter	Description	Min Value	Max Value	Unit Symbol/Notation
P_Temp	Product Temperature	−30.8	5.7	°C
E_Temp	Evaporator Temperature	−37.8	18.2	°C
C_Temp	Condenser Temperature	15.8	36.7	°C
Pow_Sup	Power Supply	134	146	V
IPC	Instant Power Consumption	0	661	W
SI	Signal	−113	85	dBm
DAS	Door Alert Status	0	1	N/A [Binary]

Table A2. Cont.

Parameter	Description	Min Value	Max Value	Unit Symbol/Notation
DCS	Door Close Status	0	1	N/A [Binary]
DOS	Door Open Status	0	1	N/A [Binary]
MCS	Machine Cooling Status	0	1	N/A [Binary]
MDS	Machine Defrost Status	0	1	N/A [Binary]
MPS	Machine Pause Status	0	1	N/A [Binary]
HPTA	High Product Temperature Alarm	0	1	N/A [Binary]

Table A3. Monitored variables for the nitrogen generation process.

Parameter	Description	Min Value	Max Value	Unit Symbol/Notation
CMS_AP	Carbon Molecular Sieve Air Pressure	0	9.5	bar
OC	Oxygen Concentration	5	500	ppm
NP	Nitrogen Pressure	0	9	bar
OOT	Oxygen Over Threshold	0	1	N/A [Binary]
CMS_PF	Carbon Molecular Sieve Pressurization Fail	0	1	N/A [Binary]

References

- Ren, L.; Zhang, L.; Wang, L.; Tao, F.; Chai, X. Cloud manufacturing: Key characteristics and applications. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 501–515. [\[CrossRef\]](#)
- Caterino, M.; Rinaldi, M.; Di Pasquale, V.; Greco, A.; Miranda, S.; Macchiaroli, R. A human error analysis in human–robot interaction contexts: Evidence from an empirical study. *Machines* **2023**, *11*, 670. [\[CrossRef\]](#)
- Ye, A.; Cai, J.; Yang, Z.; Deng, Y.; Li, X. The impact of intelligent logistics on logistics performance improvement. *Sustainability* **2025**, *17*, 659. [\[CrossRef\]](#)
- Alam, M.; Islam, M.R.; Shil, S.K. AI-Based Predictive Maintenance for US Manufacturing: Reducing Downtime and Increasing Productivity. *Int. J. Adv. Eng. Technol. Innov.* **2023**, *1*, 541–567.
- Manco, P.; Rinaldi, M.; Caterino, M.; Fera, M.; Macchiaroli, R. Maintenance Management for Geographically Distributed Assets: A Criticality-Based Approach. *Reliab. Eng. Syst. Saf.* **2022**, *218*, 108148. [\[CrossRef\]](#)
- Selcuk, S. Predictive Maintenance, Its Implementation and Latest Trends. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2017**, *231*, 1670–1679. [\[CrossRef\]](#)
- Molęda, M.; Małysiak-Mrozek, B.; Ding, W.; Sunderam, V.; Mrozek, D. From Corrective to Predictive Maintenance—A Review of Maintenance Approaches for the Power Industry. *Sensors* **2023**, *23*, 5970. [\[CrossRef\]](#)
- Florian, E.; Sgarbossa, F.; Zennaro, I. Machine Learning-Based Predictive Maintenance: A Cost-Oriented Model for Implementation. *Int. J. Prod. Econ.* **2021**, *236*, 108114. [\[CrossRef\]](#)
- Liu, Y.; Yu, W.; Rahayu, W.; Dillon, T. An Evaluative Study on IoT Ecosystem for Smart Predictive Maintenance (IoT-SPM) in Manufacturing: Multiview Requirements and Data Quality. *IEEE Internet Things J.* **2023**, *10*, 11160–11184. [\[CrossRef\]](#)
- Ledmaoui, Y.; El Maghraoui, A.; El Aroussi, M.; Saadane, R. Review of Recent Advances in Predictive Maintenance and Cybersecurity for Solar Plants. *Sensors* **2025**, *25*, 206. [\[CrossRef\]](#)
- Tsallis, C.; Papageorgas, P.; Piromalis, D.; Munteanu, R.A. Application-Wise Review of Machine Learning-Based Predictive Maintenance: Trends, Challenges, and Future Directions. *Appl. Sci.* **2025**, *15*, 4898. [\[CrossRef\]](#)
- Li, Z.; He, Q.; Li, J. A Survey of Deep Learning-Driven Architecture for Predictive Maintenance. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108285. [\[CrossRef\]](#)
- Leukel, J.; González, J.; Riekert, M. Machine Learning-Based Failure Prediction in Industrial Maintenance: Improving Performance by Sliding Window Selection. *Int. J. Qual. Reliab. Manag.* **2023**, *40*, 1449–1462. [\[CrossRef\]](#)
- Pinciroli Vago, N.O.; Forbicini, F.; Fraternali, P. Predicting Machine Failures from Multivariate Time Series: An Industrial Case Study. *Machines* **2024**, *12*, 357. [\[CrossRef\]](#)
- Salfner, F.; Lenk, M.; Malek, M. A Survey of Online Failure Prediction Methods. *ACM Comput. Surv.* **2010**, *42*, 10. [\[CrossRef\]](#)
- Polverino, L.; Abbate, R.; Manco, P.; Perfetto, D.; Caputo, F.; Macchiaroli, R.; Caterino, M. Machine Learning for Prognostics and Health Management of Industrial Mechanical Systems and Equipment: A Systematic Literature Review. *Int. J. Eng. Bus. Manag.* **2023**, *15*, 18479790231186848. [\[CrossRef\]](#)

17. Achouch, M.; Dimitrova, M.; Ziane, K.; Sattarpanah Karganroudi, S.; Dhouib, R.; Ibrahim, H.; Adda, M. On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. *Appl. Sci.* **2022**, *12*, 8081. [\[CrossRef\]](#)
18. Ahmad, W.; Khan, S.A.; Islam, M.M.M.; Kim, J.M. A Reliable Technique for Remaining Useful Life Estimation of Rolling Element Bearings Using Dynamic Regression Models. *Reliab. Eng. Syst. Saf.* **2019**, *184*, 67–76. [\[CrossRef\]](#)
19. Kumar, A.; Chinnam, R.B.; Tseng, F. An HMM and Polynomial Regression Based Approach for Remaining Useful Life and Health State Estimation of Cutting Tools. *Comput. Ind. Eng.* **2019**, *128*, 1008–1014. [\[CrossRef\]](#)
20. Kang, Z.; Catal, C.; Tekinerdogan, B. Remaining Useful Life (RUL) Prediction of Equipment in Production Lines Using Artificial Neural Networks. *Sensors* **2021**, *21*, 932. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Aydemir, G.; Acar, B. Anomaly Monitoring Improves Remaining Useful Life Estimation of Industrial Machinery. *J. Manuf. Syst.* **2020**, *56*, 463–469. [\[CrossRef\]](#)
22. Potharaju, S.; Tirandasu, R.K.; Tambe, S.N.; Jadhav, D.B.; Kumar, D.A.; Amiripalli, S.S. A Two-Step Machine Learning Approach for Predictive Maintenance and Anomaly Detection in Environmental Sensor Systems. *MethodsX* **2025**, *14*, 103181. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Senthil, C.; Sudhakara Pandian, R. Proactive Maintenance Model Using Reinforcement Learning Algorithm in Rubber Industry. *Processes* **2022**, *10*, 371. [\[CrossRef\]](#)
24. Wang, H.; Lu, W.; Tang, S.; Song, Y. Predict Industrial Equipment Failure with Time Windows and Transfer Learning. *Appl. Intell.* **2022**, *52*, 2346–2358. [\[CrossRef\]](#)
25. Søndergaard, H.A.N.; Shaker, H.R.; Jørgensen, B.N. Enabling Alarm-Based Fault Prediction for Smart Meters in District Heating Systems: A Danish Case Study. *Smart Cities* **2024**, *7*, 1126–1148. [\[CrossRef\]](#)
26. Zhang, X.; Xiao, L.; Kang, J. Degradation Prediction Model Based on a Neural Network with Dynamic Windows. *Sensors* **2015**, *15*, 6996–7015. [\[CrossRef\]](#)
27. Carrasco, J.; López, D.; Aguilera-Martos, I.; García-Gil, D.; Markova, I.; Garcia-Barzana, M.; Arias-Rodil, M.; Luengo, J.; Herrera, F. Anomaly Detection in Predictive Maintenance: A New Evaluation Framework for Temporal Unsupervised Anomaly Detection Algorithms. *Neurocomputing* **2021**, *462*, 440–452. [\[CrossRef\]](#)
28. Shukla, K.; Nefti-Meziani, S.; Davis, S. A Heuristic Approach on Predictive Maintenance Techniques: Limitations and Scope. *Adv. Mech. Eng.* **2022**, *14*, 16878132221101009. [\[CrossRef\]](#)
29. An, D.; Kim, N.H.; Choi, J.H. Practical Options for Selecting Data-Driven or Physics-Based Prognostics Algorithms with Reviews. *Reliab. Eng. Syst. Saf.* **2015**, *133*, 223–236. [\[CrossRef\]](#)
30. de Michalski, M.A.C.; Murad, C.A.; Kashiwagi, F.N.; de Souza, G.F.M.; da Silva, H.J.B.; Côrtes, H.M. A Multi-Criteria Framework for Selecting Machine Learning Techniques for Industrial Fault Prognosis. *IEEE Access* **2025**, *13*, 154508–154544. [\[CrossRef\]](#)
31. Makridis, G.; Kyriazis, D.; Plitsos, S. Predictive Maintenance Leveraging Machine Learning for Time-Series Forecasting in the Maritime Industry. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–8. [\[CrossRef\]](#)
32. Giannoulidis, A.; Gounaris, A.; Nikolaidis, N.; Naskos, A.; Caljouw, D. Investigating Thresholding Techniques in a Real Predictive Maintenance Scenario. *ACM SIGKDD Explor. Newsl.* **2022**, *24*, 86–95. [\[CrossRef\]](#)
33. Tong, S.; Yuan, J. Efficiently Enhancing Long-Term Series Forecasting via Ultra-Long Lookback Windows. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 20912–20920. [\[CrossRef\]](#)
34. Monteil, J.; Dekusar, A.; Gambella, C.; Lassoued, Y.; Mevissen, M. On Model Selection for Scalable Time Series Forecasting in Transport Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6699–6708. [\[CrossRef\]](#)
35. Cannarile, F.; Compare, M.; Baraldi, P.; Diodati, G.; Quaranta, V.; Zio, E. Elastic Net Multinomial Logistic Regression for Fault Diagnostics of On-Board Aeronautical Systems. *Aerosp. Sci. Technol.* **2019**, *94*, 105392. [\[CrossRef\]](#)
36. Naik, B.; Bhoi, G.; Sahu, R.K.; Raju, V.A.G.; Nayak, J.; Mishra, M. Optimal Light Gradient Boosting Model-Based Prognostic Approach for Remaining Useful Battery Life Prediction. *Int. J. Adv. Mechatron. Syst.* **2025**, *12*, 170–185. [\[CrossRef\]](#)
37. Li, S.; Jin, N.; Dogani, A.; Yang, Y.; Zhang, M.; Gu, X. Enhancing LightGBM for Industrial Fault Warning: An Innovative Hybrid Algorithm. *Processes* **2024**, *12*, 221. [\[CrossRef\]](#)
38. Efeoglu, E.; Tuna, G. Machine Learning for Predictive Maintenance: Support Vector Machines and Different Kernel Functions. *J. Mach. Manuf. Reliab.* **2022**, *51*, 447–456. [\[CrossRef\]](#)
39. Gao, D.; Huang, M. Prediction of Remaining Useful Life of Lithium-Ion Battery Based on Multi-Kernel Support Vector Machine with Particle Swarm Optimization. *J. Power Electron.* **2017**, *17*, 1288–1297. [\[CrossRef\]](#)
40. Li, Z.; Jiang, Y.; Duan, Z.; Peng, Z. A New Swarm Intelligence Optimized Multiclass Multi-Kernel Relevant Vector Machine: An Experimental Analysis in Failure Diagnostics of Diesel Engines. *Struct. Health Monit.* **2018**, *17*, 1503–1519. [\[CrossRef\]](#)
41. Kong, Z.; Cui, Y.; Xia, Z.; Lv, H. Convolution and Long Short-Term Memory Hybrid Deep Neural Networks for Remaining Useful Life Prognostics. *Appl. Sci.* **2019**, *9*, 4156. [\[CrossRef\]](#)
42. Wahid, A.; Breslin, J.G.; Intizar, M.A. Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework. *Appl. Sci.* **2022**, *12*, 4221. [\[CrossRef\]](#)

43. Jiang, J.R.; Lee, J.E.; Zeng, Y.M. Time Series Multiple Channel Convolutional Neural Network with Attention-Based Long Short-Term Memory for Predicting Bearing Remaining Useful Life. *Sensors* **2020**, *20*, 166. [CrossRef] [PubMed]
44. Li, W.; Li, T. Comparison of Deep Learning Models for Predictive Maintenance in Industrial Manufacturing Systems Using Sensor Data. *Sci. Rep.* **2025**, *15*, 23545. [CrossRef]
45. Al Hamed, M.; Nemer, M.A.; Azar, J.; Makhoul, A.; Bourgeois, J. Evaluating Transformer Architectures for Fault Detection in Industry 4.0: A Multivariate Time Series Classification Approach. In Proceedings of the 2025 5th IEEE Middle East and North Africa Communications Conference (MENACOMM), Byblos, Lebanon, 20–22 February 2025; pp. 1–6. [CrossRef]
46. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
47. Ismail Fawaz, H.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Deep Learning for Time Series Classification: A Review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [CrossRef]
48. Saito, T.; Rehmsmeier, M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE* **2015**, *10*, e0118432. [CrossRef]
49. Scikit-Learn Developers. sklearn.preprocessing.MinMaxScaler. Scikit-Learn Documentation. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html> (accessed on 15 May 2026).
50. Moghaddam, V.H.; Hamidzadeh, J. New Hermite Orthogonal Polynomial Kernel and Combined Kernels in Support Vector Machine Classifier. *Pattern Recognit.* **2016**, *60*, 921–935. [CrossRef]
51. Defazio, A.; Bach, F.; Lacoste-Julien, S. SAGA: A Fast Incremental Gradient Method with Support for Non-Strongly Convex Composite Objectives. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1646–1654. [CrossRef]
52. Scikit-Learn Developers. sklearn.linear_model.LogisticRegression. Scikit-Learn Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (accessed on 15 May 2026).
53. Scikit-Learn Developers. sklearn.svm.SVC. Scikit-Learn Documentation. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html> (accessed on 15 May 2026).
54. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631. [CrossRef]
55. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. [CrossRef]
56. Scikit-Learn Developers. sklearn.metrics.f1_score. Scikit-Learn Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (accessed on 15 May 2026).
57. Keles, F.D.; Wijewardena, P.M.; Hegde, C. On the Computational Complexity of Self-Attention. *arXiv* **2022**, arXiv:2209.04881. [CrossRef]
58. Proto, S.; Ventura, F.; Apiletti, D.; Cerquitelli, T.; Baralis, E.; Macii, E.; Macii, A. PREMISES, a Scalable Data-Driven Service to Predict Alarms in Slowly-Degrading Multi-Cycle Industrial Processes. In Proceedings of the 2019 IEEE International Congress on Big Data (BigDataCongress), Milan, Italy, 8–13 July 2019; pp. 139–143. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.