machines

MDPI

Article

Predicting Machine Failures from Multivariate Time Series: An Industrial Case Study

Nicolò Oreste Pinciroli Vago, Francesca Forbicini and Piero Fraternali *

Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milano, Italy; nicolooreste.pinciroli@polimi.it (N.O.P.V.); francesca.forbicini@mail.polimi.it (F.F.)
* Correspondence: piero.fraternali@polimi.it

Abstract: Non-neural machine learning (ML) and deep learning (DL) are used to predict system failures in industrial maintenance. However, only a few studies have assessed the effect of varying the amount of past data used to make a prediction and the extension in the future of the forecast. This study evaluates the impact of the size of the reading window and of the prediction window on the performances of models trained to forecast failures in three datasets of (1) an industrial wrapping machine working in discrete sessions, (2) an industrial blood refrigerator working continuously, and (3) a nitrogen generator working continuously. A binary classification task assigns the positive label to the prediction window based on the probability of a failure to occur in such an interval. Six algorithms (logistic regression, random forest, support vector machine, LSTM, ConvLSTM, and Transformers) are compared on multivariate time series. The dimension of the prediction windows plays a crucial role and the results highlight the effectiveness of DL approaches in classifying data with diverse time-dependent patterns preceding a failure and the effectiveness of ML approaches in classifying similar and repetitive patterns preceding a failure.

Keywords: failure prediction; machine learning; deep learning; predictive maintenance; compressor; blood refrigerator; nitrogen generator; wrapping machine

Citation: Pinciroli Vago, N.O.; Forbicini, F.; Fraternali, P. Predicting Machine Failures from Multivariate Time Series: An Industrial Case Study. *Machines* **2024**, *12*, 357. https://doi.org/10.3390/machines12060357

Academic Editors: Hongli Gao and Zhichao You

Received: 12 April 2024
Revised: 16 May 2024
Accepted: 20 May 2024
Published: 22 May 2024

1. Introduction

Predictive maintenance is a strategy that aims to reduce equipment degradation and prevent failures [1–3] by anticipating their occurrence. It differs from corrective maintenance (in which the components are used for their entire life span and replaced when there is a fault) [4,5] and preventive maintenance (in which the replacement and reparation activities are scheduled periodically) [6,7].

Industrial data are often the result of monitoring several physical variables, which produces multivariate time series [8,9]. Predicting failures via manual inspection is time-consuming and costly, mainly because the correlation among multiple variables can be difficult to appraise with manual approaches. Computer-aided methods can improve performance and reliability and are less onerous and error-prone [10–12].

Predicting machine failures requires the cycle of data collection, model identification, parameter estimation, and validation [13]. Alternative models are compared in terms of the effort needed to train them, the performances they deliver, and the relationship between such performances and the amount of input data used to make a prediction and the time horizon of the forecast.

In this paper, we compare alternative models trained for predicting malfunctions. The focus of the analysis is to study how performances, measured with the macro F_1 score metrics, are affected by the length of the input time series fragment used to make the prediction (*reading window—RW*) and by the extension in the future of the forecast (*prediction window—PW*).

As case studies, we consider the time series made of multiple telemetry variables collected with Internet of Things (IoT) sensors from three machines: (1) an industrial

wrapping machine, (2) a blood refrigerator, and (3) a nitrogen generator. The IoT method used for data collection is stream IoT (i.e., data are collected continuously) [14] and the protocol is Message Queuing Telemetry Transport (MQTT).

The three datasets are created using IoT sensors that collect physical quantities to monitor the system status. The emission of alert codes by the machines' hardware signals the occurrence of anomalous conditions. The goal of the described data-driven predictive maintenance scenario is to anticipate the occurrence of a fault (i.e., a malfunctioning indicated by an alert code) within a future time interval (i.e., the prediction window) using historical data (i.e., the reading window). Labelled data (i.e., the actual alert codes emitted by the machine in past work sessions) make it possible to formulate the predictive maintenance problem as a binary classification task addressed with a fully supervised approach.

Our experimental setting contrasts non-neural machine learning (ML) and deep learning (DL), which are popular methods for addressing predictive maintenance tasks. ML methods have been widely used in diverse applications [12,15–18] and recent works also tested DL methods [19,20]. A few works have compared supervised ML and DL methods for failure prediction in time series data [20,21] and evaluated the influence of either the reading window (e.g., [17,18,22]) or of the prediction window (e.g., [16,21,23–25]). Only a few consider both parameters [12,26].

The contribution of the paper can be summarized as follows:

- We compare three ML methods (logistic regression, random forest, and Support Vector Machine) and three DL methods (LSTM, ConvLSTM, and Transformer) on three novel industrial datasets with multiple telemetry data.
- We study the effect of varying the size of both the reading window and the prediction window in the context of failure prediction and discuss the consequences of these choices on the performances.
- We define and evaluate the diversity of patterns preceding faults and the datasets' complexities, showing that DL approaches outperform ML approaches significantly only for complex datasets with more diverse patterns.
- We show that all methods lose predictive power when the horizon enlarges because the temporal correlation between the input and the predicted event tends to vanish.
- We highlight that when patterns are diverse, the amount of historical data becomes influential. However, augmenting the amount of input is not beneficial in general.
- We publish the datasets and the compared algorithms (available at https://github.com/nicolopinci/polimi_failure_prediction (accessed on 19 May 2024)). The published datasets are among the few publicly accessible by researchers in the domain of fault prediction for industrial machines [27].

The remainder of this paper is structured as follows: Section 2 surveys the related work, Section 3 presents the methodology adopted for pre-processing the data and making predictions, Section 4 presents and discusses the results, and Section 5 presents the conclusions and proposes future work.

2. Related Work

Anomaly detection and failure prediction are two related fields applied to diverse data, including time series [28–32] and images [33–35]. This research focuses on failure prediction applied to multivariate time series data acquired by multiple IoT sensors connected to an industrial machine. Several works have surveyed failure prediction on time series [31,36,37].

Failure prediction approaches belong to three categories, depending on the availability and use of data labels: supervised [12,19], semi-supervised [38], and unsupervised [39]. Predicting failures in a future interval given labelled past data can be considered a supervised binary classification task [12]. The dataset used in this research is labelled, making supervised methods a viable option.

ML and DL are two popular approaches for failure prediction in time series data. For supervised failure prediction, ML methods include decision trees (DT) [15,16], gradient

boosting (GB) [16,17], random forests (RF) [15–18], support vector machines (SVM) [15,16], and logistic regression (LR) [12,18]. DL methods include neural network architectures such as recurrent neural networks (RNNs) [20] and convolutional neural networks (CNNs) [20]. Some works have compared supervised ML and DL methods for failure prediction in time series data [20,21].

RF is one of the most used approaches [12,15–18,22,23,25,30,40–45]. The work in [25] compares RF with ML and DL approaches, including Neural Networks and SVM, and shows the superiority of RF in terms of accuracy for a prediction window not longer than 300 s for a case study of wind turbines. However, accuracy does not consider class unbalance and is not appropriate for evaluating the performance of failure prediction algorithms [31]. The work in [42] also considers the case of wind turbines and presents only the performances of RF using accuracy, sensitivity, and specificity for a prediction window of 1 h. The work in [22] also uses RF on wind turbine data and measures performances using precision and recall for varying the reading windows up to 42 h, showing rather poor performances. Other applications of RF in predictive maintenance range from refrigerator systems [45] to components of vending machines [43], vehicles [30], and chemical processes [17].

SVM is another approach commonly used in supervised failure prediction [12,15,16,25,26,43,46–48]. The work in [26] presents an application of SVM to railway systems and evaluates the performances using False Positive Rate (FPR) and recall. It is one of the few approaches that consider variation in PW and RW. However, the assessment of the contribution of each window is not performed because they were changed together. The work in [16] compares ML algorithms (DT, RF, GB, and SVM) and shows that GB outperforms the other approaches, followed by RF. The work in [15] compares the accuracy of six ML algorithms on industrial data, including SVM, RF, and DT. It shows that DT not only surpasses RF but is also more interpretable. However, precision and recall would be necessary to evaluate the quality of predictions, as the dataset is likely unbalanced. The work in [18] also measures performances using accuracy (for a balanced dataset), compares RF, LR, and Ctree, and shows that (1) RF outperforms the other algorithms and (2) the reading window size does not have a significant impact on the performances.

A few works contrast ML and DL predictors. The work in [21] compares the performances using Area Under the Curve (AUC) and shows that a small fully-connected neural network outperforms Naïve Bayes. However, other popular methods (e.g., SVM, LR, LSTM) are not evaluated. The work in [20] is one of the few approaches considering a comparison between two DL approaches (a CNN and LSTM) and shows (1) the superiority of the convolutional network in terms of accuracy and (2) the decrease in accuracy as the prediction window increases. However, additional metrics would be necessary to evaluate performances on a likely unbalanced dataset.

Considering alternative DL approaches, aside from the commonly employed LSTM-based architectures [49–51], there is noteworthy research interest in ConvLSTM models. The work in [52] finds that ConvLSTM performs similarly to random forest (RF) when predicting faults in wind turbine gearbox systems. The work in [53] shows the effectiveness of a ConvLSTM-based architecture for failure detection. Both works do not analyze the impact of the size of RW and PW nor compare alternative DL architectures. ConvLSTM has also proved effective in forecasting the value of telemetry variables, a preliminary step for fault prediction in diverse systems [54–56]. Transformer-based architectures have been applied to anomaly detection [57,58], and fault diagnosis [59,60]. The work in [61] shows the effectiveness of a Transformer-based architecture for the fault prediction task in an Electric Power Communication Network. In contrast to LSTM-based architectures, Transformers can capture long-range dependencies and offer parallel processing capabilities, which also makes them suitable for time series tasks over long intervals.

Several works investigate the failure prediction task by forecasting the occurrence of a failure within a pre-determined time frame [17,18,21,26]. Some studies evaluate the impact

of either the reading window length [17,18,26] or the prediction window length [21,23,24]. Only a few [12,26] evaluate the impact of varying the reading and the prediction window. Considering the surveyed approaches, the work in [12] is the only one that proposes a comparison between different ML algorithms varying both the RW and PW and assessing the contribution of each. However, the comparison does not consider DL algorithms.

In summary, a few works on supervised failure prediction for multivariate time series evaluate the variation of both RW and PW. However, none compare ML and DL algorithms in this respect. Moreover, none of the surveyed papers consider the presence of discrete sessions in the telemetry time series, which is instead the case of the wrapping machine dataset. Furthermore, the related works very rarely disclose the industrial datasets used in the experiments, which hinders the reproducibility of research and, ultimately, the progress of the status of the art.

Our work investigates the contribution of both the RW and PW variations and compares diverse ML and DL approaches on three novel industrial datasets featuring both discrete session-based and continuous time machines. The collected datasets, which derive from the telemetry of real machines and comprise hardware-produced failure alerts as ground truth, are available in a public repository.

3. Materials and Methods

This section outlines the experimental design employed to evaluate the impact of the reading/prediction windows and of the hyperparameters on the performance of ML and DL approaches. It focuses on the materials (the datasets, presented in Section 3.1) and the methods (the data pre-processing procedures, the experimental methodology and the performance metrics, presented in Sections 3.2–3.6).

3.1. Datasets

This section presents the three datasets used in this research. The datasets have been acquired from an industrial company operating in the remote control of industrial and domestic appliances and have been selected based on their different complexities (defined using spectral entropy, as detailed in Section 3.2) and diverse dynamics and applications. Wrappers represent machines that operate in discrete sessions. Blood refrigerators display quasi-periodic dynamics over a limited period (in the order of tens of minutes). Nitrogen generators exhibit slow dynamics with regular patterns that repeat over extended periods (in the order of several hours).

3.1.1. Wrapping Machine

Wrapping machines are systems used for packaging and comprise components such as motors, sensors, and controllers. The monitored exemplars are semi-automatic machines that wrap objects carried on pallets with stretch film. Each machine comprises the *turntable*, a central circular platform for loading objects; the *lifter*, a moving part that contains the wrapping film reel; the *tower*, a column on which the lifter moves; the *platform motor*, a gear motor controlling the movements of the turntable with a chain drive, and the *lifting motor*, an electric motor for lifting the carriage.

The functioning of the machine consists of cycles with irregular lengths, depending on the size of the wrapped objects. A cycle consists of five steps:

1. Loading the products on a pallet at the centre of the rotating platform.
2. Tying the film's trailing end to the pallet's base.
3. Starting the wrapping cycle, executed according to previously set parameters.
4. Manually cutting the film (through the cutter or with an external tool) and making it adhere to the wrapped products.
5. Removing the wrapped objects.

The dataset contains records from sensors installed on a wrapping machine collected over one year (1 June 2021 to 31 May 2022). The machine is equipped with sensors attached to different components, such as the rotating platform and the electric motors.

An acquisition system gathers data from sensors and sends them to a storage server. Sensors measure over 100 quantities (e.g., temperatures, motor frequencies, and platform speeds). Each data item records a timestamp and the last value measured by a sensor. Data are transmitted only when at least one sensor registers a variation for saving energy, resulting in variable data acquisition frequency. Data are grouped into work sessions that represent periods in which the machine is operating. Each work session has a starting point, manually set by the operator, corresponding to the start of an interval during which the machine wraps at least one pallet. A work session ends when the machine finishes wrapping the pallets or when a fatal alert occurs. Session-level metrics are collected, too, such as the number of completed pallets and the quantity of consumed film. The dataset contains information about alerts thrown by the machine during the power-on state (i.e., when the machine is powered on, which does not necessarily mean it is producing pallets). An alert is a numerical code that refers to a specific problem that impacts the functioning of the machine.

Depending on the changes detected by sensors, the irregular sampling time is addressed by resampling the time series with a frequency of 5 s, which gives a good trade-off between memory occupation and signal resolution. Missing values are filled in using the last available observation because the absence of new measurements indicates no change in the recorded value.

3.1.2. Blood Refrigerator

Blood refrigerators are systems designed to store blood safely and its derivatives at specific temperatures. Each unit includes different components. *Compressor*: pumps the refrigerant, increasing its pressure and temperature. *Condenser*: cools down and condenses the refrigerant. *Evaporator*: causes the refrigerant to evaporate the inside of the unit for cooling. *Expansion Valve*: regulates the flow of refrigerant into the evaporator. *Interconnecting Tubing*: facilitates the movement of the refrigerant between the compressor, condenser, expansion valve and evaporator.

A refrigeration cycle lasts ≈10 min and comprises the following phases:

1. Compression: The cycle begins with the compressor drawing in low-pressure, low-temperature refrigerant gas. The compressor then compresses the gas, which raises its pressure and temperature.
2. Condensation: The high-pressure, high-temperature gas exits the compressor and enters the condenser coils. As the gas flows through these coils, it releases heat, cools down, and condenses into a high-pressure liquid.
3. Expansion: This high-pressure liquid then flows through the capillary tube or expansion valve. As it does, its pressure drops, causing a significant drop in temperature. The refrigerant exits this stage as a low-pressure, cool liquid.
4. Evaporation: The low-pressure, cool liquid refrigerant enters the evaporator coils inside the refrigerator. Here, it absorbs heat from the interior, causing it to evaporate and turn back into a low-pressure gas. This process cools the interior of the refrigerator.
5. Return to Compressor: The low-pressure gas then returns to the compressor and the cycle starts over.

The dataset includes information from IoT sensors that measure 58 variables (e.g., the status of the door of the refrigerator, its temperature, and energy consumption). The series comprises data from 31 October 2022 to 30 December 2022 and records only when a variable changes value. This results in an irregular sampling period varying from a few seconds to one hour. For this reason, we resampled the dataset using the median of the original sampling frequency (i.e., ≈34 s).

3.1.3. Nitrogen Generator

Nitrogen generators separate the nitrogen from the other gasses, such as oxygen, in compressed air. They include: an *air compressor*, to supply the compressed air; *Carbon Molecular Sieves (CMSs)*, to filter other gasses from the nitrogen; *absorption vessels*, to take the

nitrogen not filtered out by the CMS; *towers*, to increase the nitrogen production. A tower comprises one CMS, two or more absorption vessels, and the space where the division between nitrogen and oxygen takes place; *valves*, to direct the flow of air and nitrogen and regulate the absorption of the nitrogen by the vessels; a *Buffer Tank*, to store the purified nitrogen.

The generator works as follows:

1. The air compressor compresses the air to a high pressure and supplies it to the machine.
2. In the towers, the CMS adsorbs smaller gas molecules, such as oxygen, while allowing larger nitrogen molecules to pass through the sieve and go into the vessels.
3. The buffer tank receives the nitrogen gas from the vessels through the valves.
4. The valves reduce the pressure in the current working tower to release the residual gasses, whereas in the other towers, the pressure is increased to restart the process.

Each phase lasts $\approx$ 8 h, but the CMS air pressure reduces only in the last $\approx$ 2.5 h.

The dataset includes information from IoT sensors installed in the generator. A set of 64 variables measures the essential work parameters (e.g., the nitrogen and the CMS air pressure). The series comprises data from 1 August 2023 to 29 September 2023 and contains values recorded when a change in a variable occurs. This results in an irregular sampling period varying from a few seconds to four hours. For this reason, we resampled the dataset using 1 min as frequency.

3.2. Data Processing

Data processing is an essential step in analyzing each dataset, involving selecting pertinent variables associated with physical quantities and the resampling period. The anomalous patterns preceding faults can be considered to characterize the diversity of the datasets. To this end, the dataset is first normalized using min-max normalization, and then a set of fixed-size windows preceding each fault is defined. For every pair of such windows, denoted as w_i and w_j, the Euclidean distance, d_{ijkv}, is calculated between pairs of data points at corresponding positions, indexed by k, for each variable, represented as v. The distance is computed between the points p_{ikv} and p_{jkv}. The diversity of patterns is quantified as the mean value scaled in the range $[0, 1]$:

$$m = mean_{ijkv}(d_{ijkv}) \tag{1}$$

To characterize the complexity of a dataset as a whole, we compute the mean of the spectral entropies of the considered time series [62], as implemented in the `antropy` library (https://raphaelvallat.com/antropy/build/html/generated/antropy.spectral_entropy.html (accessed on 19 May 2024)), normalized between 0 and 1 and using the fft method [63]. Higher entropy values correspond to a higher time series complexity.

3.2.1. Wrapping Machine

The first phase is feature selection. Since not all the quantities are equally relevant, only 13 are kept based on the data description provided by the manufacturer and are summarized in Table 1. Most removed features were redundant or had constant values. For example, such features as the employed recipe and the firmware version are excluded because they do not describe the system dynamics, whereas such features as the variation of a variable value during the session are eliminated because they can be derived from the initial and final values.

The second phase is selecting relevant alert codes to anticipate machine malfunction. Figure 1 presents the distribution of the alert codes most relevant according to the manufacturer and Table 2 explains their meaning. Most of them are rare (i.e., are observed less than 10 times), and alerts 11 (platform motor inverter protection) and 34 (machine in emergency conditions) are the most common ones. Alert 34 is thrown when the operator presses an emergency button. However, this occurrence heavily depends on human behaviour because the operators often use the emergency button as a quick way to turn the machine off,

as confirmed by the plant manager. For this reason, alert 34 is discarded, and the prediction task focuses only on alert 11. This alert is fundamental for the correct functioning of the machine because the inverter controls the frequency or power supplied to the motor and thus controls its rotation speed. The improper control of the rotation speed hinders the wrapping operation.

Table 1. The 13 variables obtained after pre-processing. The minimum and maximum values are reported for each variable and the measurement unit. The "Movement" column indicates whether the variable is used to determine whether any motors are moving.

Variable Name	Minimum	Maximum	Unit	Movement
Platform motor frequency	0	5200	Hz	✓
Current speed cart	0	100	%	✓
Platform motor speed	0	100	%	✓
Lifting motor speed	0	88	RPM (Revolutions per Minute)	✓
Platform rotation speed	0	184	RPM	✓
Slave rotation speed	0	184	m/min	✓
Lifting speed rotation	0	73	m/min	✓
Flag roping	0	31	µm	
Platform position	0	359	°	
Temperature hoist drive	0	55	°C	
Temperature platform drive	0	61	°C	
Temperature slave drive	0	55	°C	
Total film tension	−100	9900	%	

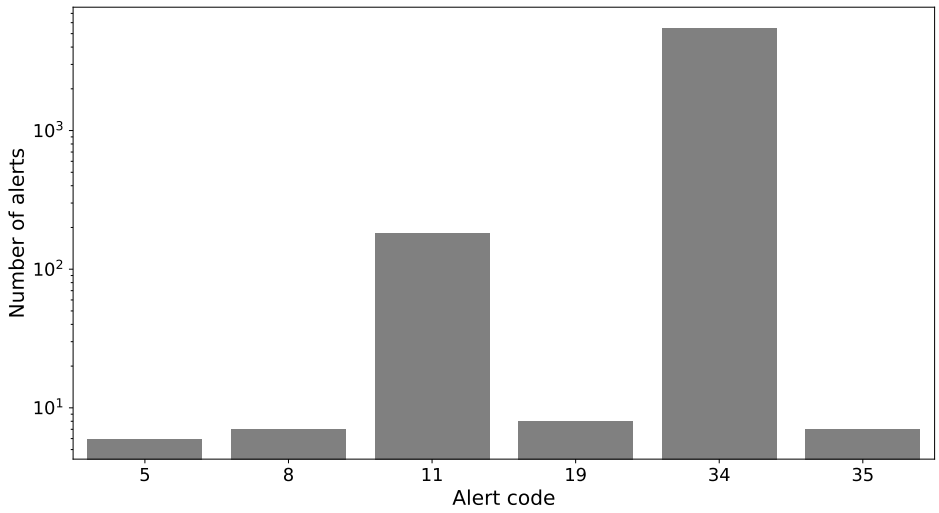


Figure 1. Distribution of the alert codes on a logarithmic scale. Alert 34 is the most frequent but must be discarded because it is unreliable. Alert 11 is the second most frequent. The remaining alerts have less than ten occurrences in the entire dataset.

Table 2. Explanation of the wrapping machine alert codes.

Alert	Description
5	Loose carriage belt alarm
8	Film end alarm
11	Platform motor inverter protection
19	Carriage motor inverter protection
34	Machine in emergency conditions
35	Table phonic wheel sensor fault

In the examined dataset, alert 11 is preceded by diverse time-dependent patterns, and the mean Euclidean distance defined in Equation (1) is ≈0.26 for a window of 15 min. Figure 2 shows two examples of patterns preceding faults. The spectral entropy is ≈0.72.

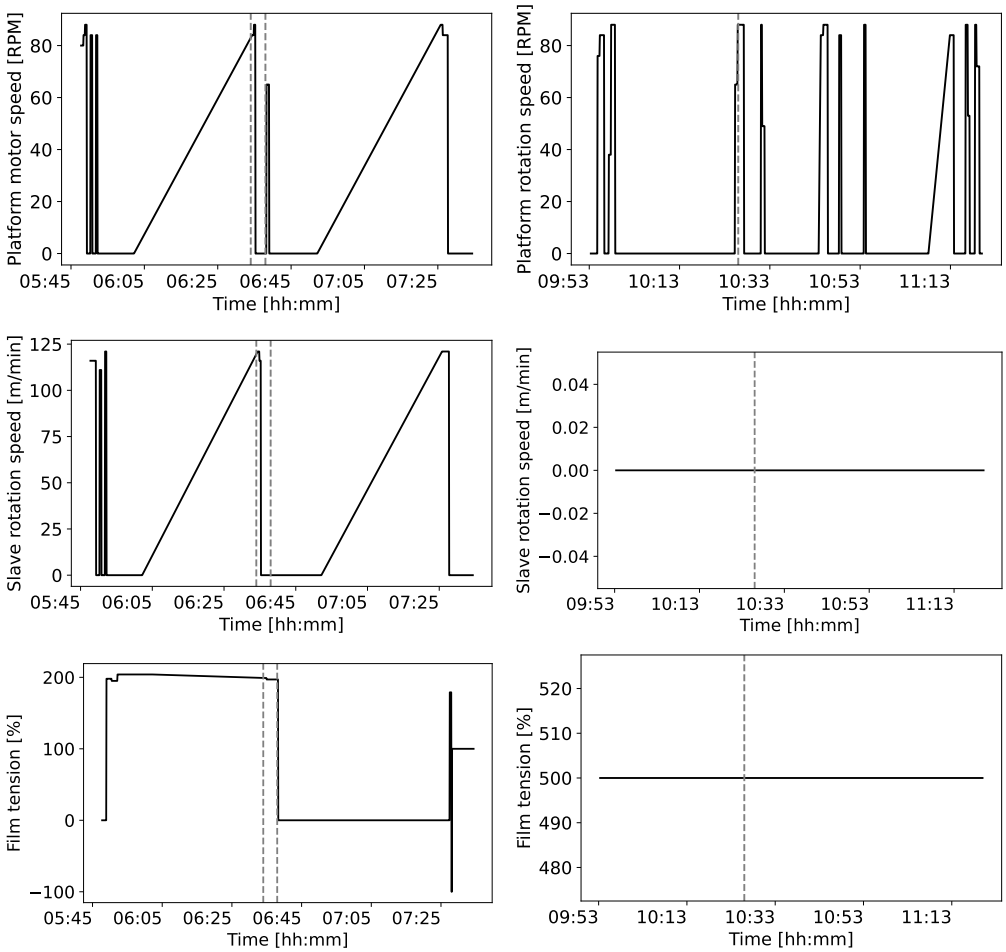


Figure 2. Examples of diverse patterns preceding faults in the wrapping machine dataset. The patterns involve three variables and the faults are displayed as dashed vertical lines. The example on the left shows two faults observed on 15 June 2021. They are preceded by a slow decrease in the film tension, and after the second fault, the slave motor has a speed of zero. On the right, the fault observed on 30 June 2021 is preceded by a sudden acceleration of the platform rotation speed, whereas the slave motor does not move, but the film tension remains constant.

The work session boundaries provided in the dataset also depend on human intervention. Some inconsistent session-level metrics and a large number of sessions that do not produce any pallets are observed. A more reliable definition of a work session can be inferred from the telemetry variables. Intuitively, a session start time is introduced when at least one motion variable increases its value, and a session end time is created when all motion variables have decreased their value in the last ten minutes. Algorithm 1 specifies how sessions are defined.

Algorithm 1 Computation of session boundaries

Require: *ds*, the dataset containing, for each timestamp, an array with the values of the movement variables, structured as a hash map with the timestamp *ts* as the key and the movement variables array *mv* as the value

Ensure: *sessions*, a map that associates a session number to each timestamp and -1 to out-of-session timestamps

```
latest ← NULL
flag ← false
sessions ← HashMap()
counter ← 1
for ts ∈ ds.timestamps do
    sessions[ts] ← −1
    if ts ≠ min(ds.timestamps) then
        pts ← max({x : x ∈ ds.timestamps and x < ts})
        δ ← ds[ts].mv − ds[pts].mv
        if ∃e ∈ δ | e > 0 then
            latest ← ts
            flag ← true
            sessions[ts] ← counter
        else
            if flag = true then
                sessions[ts] ← counter
                if ts − latest > 10 min then
                    counter ← counter + 1
                    flag ← false
                end if
            end if
        end if
    else
        latest ← ts
    end if
end for
```

Figure 3 shows the distribution of the work session duration computed from the telemetry data. The idle time intervals between sessions are neglected in further processing.

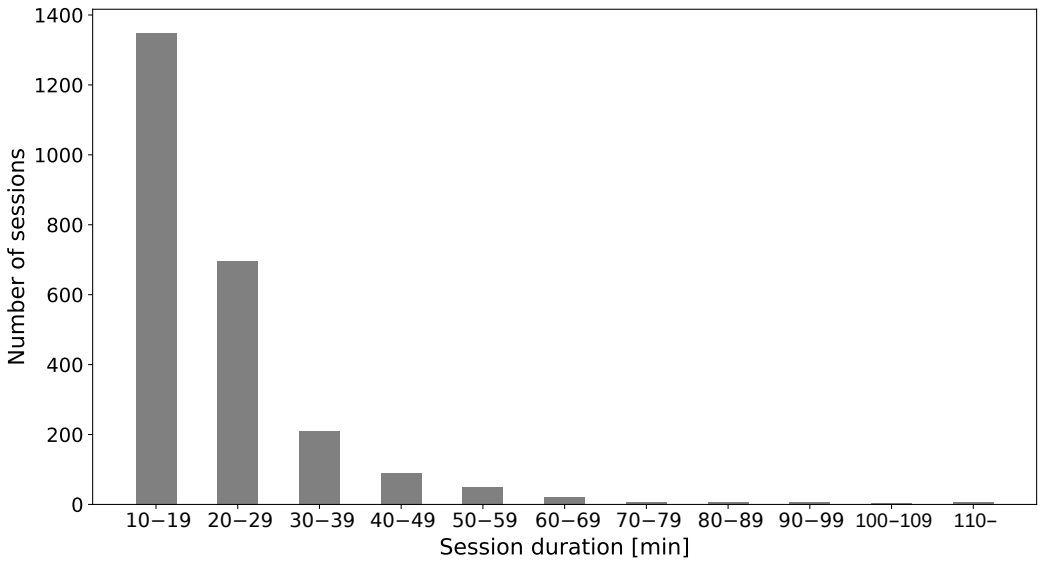


Figure 3. Distribution of the duration of work sessions.

After pre-processing, it is possible to compute the distance of each data sample within a work session to the successive alert, i.e., the Time to Failure (TTF).

3.2.2. Blood Refrigerator

The data comprise variables directly measured by sensors (base variables) and variables derived from the base ones. For instance, the base variable "Door status" represents the status of the refrigerator door and has an associated derived variable "Door opening daily counter", which counts every day when the door is open. We keep the 12 most significant variables, shown in Table 3.

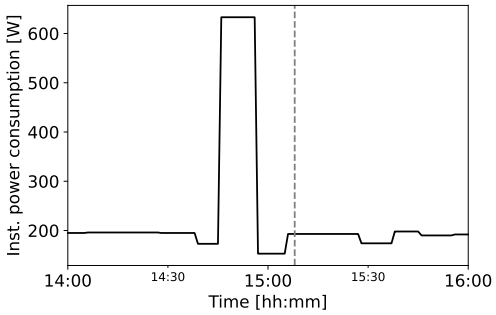
Table 3. The 12 variables of the blood refrigerator.

Variable Name	Minimum	Maximum	Unit
Product temperature Base	−30.8	5.7	°C
Evaporator temperature base	−37.8	18.2	°C
Condenser temperature base	15.8	36.7	°C
Power supply	134.0	146.0	V
Instant power consumption	0.0	661.0	W
Signal	−113.0	85.0	dBm
Door alert	0	1	
Door close	0	1	
Door open	0	1	
Machine cooling	0	1	
Machine defrost	0	1	
Machine pause	0	1	

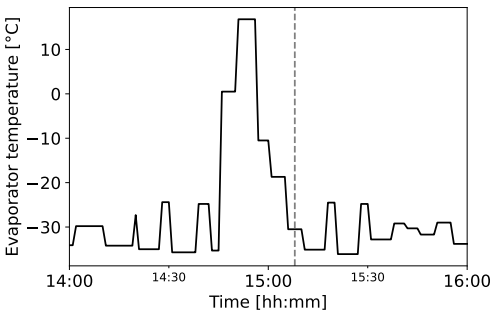
The dataset also contains alarms. Specifically, the dataset originally contained 15 types of alarms (Alarm Probe Failure, Condenser Probe Failure, Control Probe, Defrost Timeout Failure, Evaporator Probe Failure, Excessive Compressor Use Failure, High Condenser Temperature, High-Temperature Compartment, High Product Temperature, Low Evaporator Temperature, Low-Temperature Product, Protection Against Anti-Freezing, Protection Against Over-Heating, Blackout, and Device offline), but those described as critical by the manufacturer are the ones presented in Table 4. After analyzing two such alarms, we have selected "alarm 5" as the prediction target because "alarm 1" has too few occurrences, making the analysis of performances unfeasible. Storing blood at low temperatures is crucial, as higher temperatures can lead to bacterial growth [64], hemolysis [65], and increased risk of adverse reactions, including life-threatening conditions [66]. Figure 4 presents the typical anomalous pattern preceding a fault for three variables. In this case, the mean Euclidean distance defined in Equation (1) is ≈0.21 for a window of 15 min, ≈20% less than the one of the wrapping machine dataset. The spectral entropy is ≈0.59, indicating a lower complexity with respect to the wrapping machine dataset.

Table 4. The alarms of the blood refrigerator.

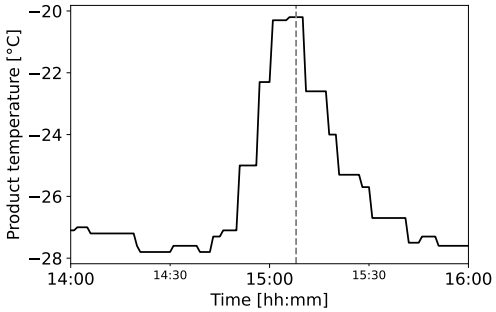
Name of the Alarm	Description	Number of Data Points
alarm 1	High-Temperature Compartment	2
alarm 5	High Product Temperature	188



(**a**) Instantaneous power consumption.



(**b**) Evaporator temperature.



(**c**) Product temperature.

Figure 4. Examples of patterns preceding a fault in a blood refrigerator, observed on 22 November 2022. Before the fault (shown by a dashed line), the instant power consumption increases and then decreases after a short time, the evaporator temperature decreases by ≈40 °C after a sudden increase, and the product temperature has an increase.

3.2.3. Nitrogen Generator

The data are collected by IoT sensors over two months (1 August 2023 to 29 September 2023). We have considered only the variables directly measured from the sensors and discarded the derived ones. Table 5 shows the relevant variables, which record the changes in the physical status of the system.

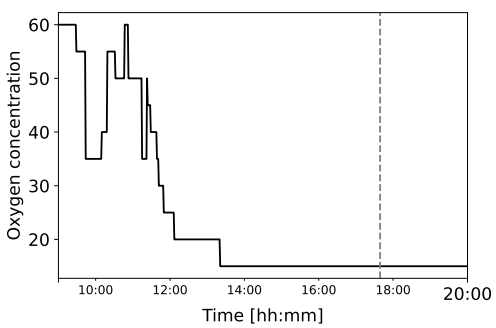
Table 5. The 4 variables of the nitrogen generator.

Variable Name	Minimum	Maximum	Unit
CMS air pressure	0.0	9.5	bar
Oxygen base concentration	5.0	500.0	
Nitrogen pressure	0.0	9.0	bar
Oxygen over threshold	0	1	

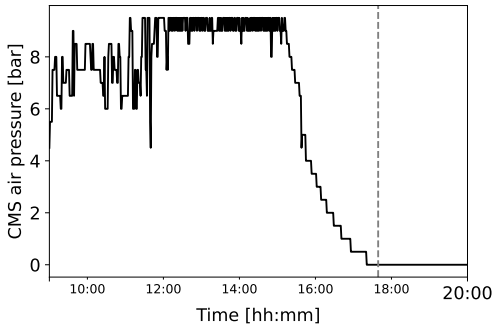
The nitrogen generator can produce different alarms, including "Air pressure too high" and "Oxygen failure—Second threshold reached". However, only one type of alarm is present in the dataset: "CMS pressurization fail", whose frequency is shown in Table 6. A failure in CMS pressurization can hinder the separation of oxygen and nitrogen, as mentioned in [67]. Figure 5 presents the typical anomalous pattern of three variables preceding a fault. The mean Euclidean distance defined in Equation (1) is ≈0.09 for a window of 15 min, ≈65% less than the one of the wrapping machine dataset, and ≈57% less than the one of the blood refrigerator dataset. The spectral entropy is ≈0.53, indicating a lower complexity with respect to both the wrapping machines and the blood refrigerator datasets.

Table 6. The alarm of the nitrogen generator.

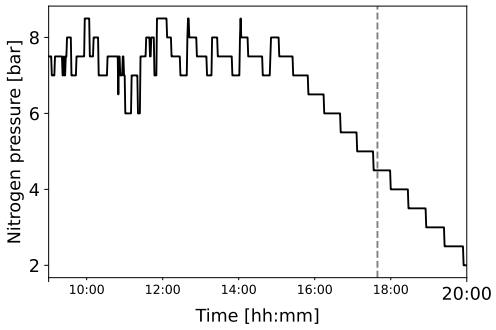
Name of the Alarm	Description	Number of Data Points
CMS pressurization fail	Pressurization of nitrogen sieve failed	54



(**a**) Oxygen concentration.



(**b**) CMS air pressure.



(**c**) Nitrogen pressure.

Figure 5. A typical pattern in a nitrogen generator, observed on 1 August 2023. In this case, before a fault (indicated with a grey dashed line), the oxygen concentration drops, the CMS air pressure decreases by ≈8 bar, and the nitrogen pressure decreases by ≈4 bar, making the nitrogen generation impossible.

3.3. Definition of the Reading and Prediction Windows

The task for comparing ML and DL methods is predicting the occurrence of alerts within a future time interval (i.e., a prediction window PW) given historical data (i.e., a reading window RW). This task is formulated as a binary classification, which assigns a label (failure/no failure) to each RW [31]. Failures were provided by the manufacturer, which measured them using sensors installed in the machines. A failure label associated with an RW is assigned when the corresponding PW contains at least an alert code. It is interpreted as a high probability of an alert occurring in the PW next to it. Given a session of length N and an RW size S_{RW}, $N - S_{RW} + 1$ reading windows are extracted. Each RW starts at timestamp t_i, with $i = 1 \dots N - S_{RW} + 1$. For an RW $(t_j \dots t_j + S_{RW})$, the corresponding PW starts at time stamp $t_j + S_{RW} + 1$. The proposed approach can be applied in the absence of sessions without loss of generality because the entire interval of the dataset can be considered as one session.

The size of RWs depends on the characteristics of the datasets. For wrapping machines, it ranges from 10 min (the minimum session duration) to 35 min. For blood refrigerators, it ranges from 10 min (a cycle duration) to 30 min. For nitrogen generators, it ranges from 10 min to 30 min because (1) the system shows minimal long-term drift, making longer historical trends irrelevant, and (2) short RWs allow for timely intervention in case of failures, without the need to collect a large amount of data before predicting a failure.

3.4. Class Unbalance

Alerts are anomalies and thus, by definition, rarer than normal behaviours [68]. For this reason, the number of RWs with the label *no failure* is significantly greater than that of RWs with the label *failure*, leading to a heavily imbalanced dataset. For all the datasets, Random UnderSampling (RUS) [69,70] is applied to balance the class distribution by making the cardinality of the majority class comparable to that of the minority class. RUS has demonstrated its effectiveness in diverse fields where the class imbalance is common, such as astrophysics [71,72], geoscience [73], and industrial informatics [74–76]. The algorithm randomly selects and removes observations from the majority class until it achieves the desired equilibrium between the two classes. In the case of the wrapping machine, RUS is applied separately on each train set (comprising four folds) and test set (one fold) for each combination of RW and PW sizes. To prevent the presence of similar data in the train and test sets (i.e., partially overlapping data), after the definition of the test set, the partially overlapping windows in the train set are neglected by RUS. The RUS step is applied 10 times, as proposed in [12]. In the blood refrigerator and nitrogen generator datasets, RUS is applied on the train and validation sets during the training phase, but not on the test set, where the test F_1 score is evaluated by assigning a different weight to the *failure* and *no failure* classes in order to account for the unbalance.

3.5. Algorithms and Hyperparameter Tuning

The algorithms for which the influence of the RW and PW size is assessed include logistic regression (LR), as a representative of the Generalized Linear Models family [77], random forest (RF), as a representative of the Ensemble Methods family [78], Support Vector Machine (SVM), as a representative of the Kernel Methods family [79], Long Short-Term Memory (LSTM) [80], Transformers [81] and ConvLSTM [54], as representatives of DL methods.

For SVM, RF, and LR, the implementation used in this work is from the Python library `scikit-learn` (https://scikit-learn.org/stable/ (accessed on 19 May 2024)), whereas DL models use `TensorFlow` (https://www.tensorflow.org/ (accessed on 19 May 2024)). The tested network architectures of the DL models and the hyperparameters resulting from the hyperparameter search of all models can be found in the project repository (available at https://github.com/nicolopinci/polimi_failure_prediction (accessed on 19 May 2024)).

For LR, C is the inverse of the regularization strength, and smaller values indicate a stronger regularization.

In the case of RF, the search varies the number of internal estimators N_E (i.e., the number of decision trees) and the maximum number of features that can be selected at each split (in this context, a feature is defined as the value of a variable at a specific timestamp).

In the case of SVM, the C coefficient is the regularization hyperparameter multiplied by the squared L_2 penalty and is inversely proportional to the strength of the regularization. Small values of C lead to a low penalty for misclassified points, whereas if C is large, SVM tries to minimize the number of misclassified examples.

Considering LSTM, in the case of the wrapping machine, the only hyperparameters that vary are the type of LSTM layer, Unidirectional or Bidirectional, and the loss function. For the latter, we compare the sigmoidF_1 loss with $\beta = 1$ and $\eta = 0$ (from now on referred to as F_1 loss) [82] and the Binary Cross-Entropy (BCE) loss. The use of BiLSTM has been shown to be effective in the works [80,83], which focus on forecasting problems. The two compared loss functions have been proven to deliver the top performances in [82]. Each loss function serves a distinct purpose and offers advantages and disadvantages. The F_1 loss considers precision and recall, thus providing a balanced model performance evaluation. In this case, minimizing the F_1 loss means maximizing the F_1 score for the "Failure" class (i.e., reducing the missed "Failure" occurrences). The BCE loss is a widely used loss function for binary classification tasks and is symmetric with respect to the two classes. It measures the dissimilarity between predicted probabilities and target labels, encouraging the model to improve its probability estimations. BCE loss encourages the model to output

probabilities for each class, providing richer information about the model's confidence in its prediction. However, it does not inherently consider the interplay between precision and recall and is not the most suitable choice for tasks where a balance between precision and recall is essential.

In the case of Transformers, the BCE loss function is employed [84], and two hyperparameters are relevant: the number of attention heads and the number of transformer blocks. The former affects the capacity of the model to capture diverse patterns and relationships in the data, with higher values improving the ability to discern patterns. The latter increases the complexity of the model, thus allowing it to capture more complex patterns while augmenting the risk of overfitting.

In the case of ConvLSTM [54], BCE is chosen as the loss function, and the hyperparameter search (described in the project repository) has varied both the kernel size and the number of filters. Increasing the kernel size extends the receptive field, and augmenting the number of filters allows for capturing more diverse and complex spatiotemporal features.

3.6. Training and Evaluation

Results are assessed using the macro F_1 score, which extends the F_1 score, defined in Equation (2). The F_1 score depends on precision and recall, defined, respectively, in Equations (3) and (4), where TP stands for the number of true positives, FP for the number of false positives, TN for the number of true negatives, and FN for the number of false negatives.

$$F_1 = 2 \cdot \frac{PREC \cdot REC}{PREC + REC} \tag{2}$$

$$PREC = \frac{TP}{TP + FP} \tag{3}$$

$$REC = \frac{TP}{TP + FN} \tag{4}$$

The macro F_1 score is the average of the F_1 scores computed for each class independently. To compute the F_1 score for each class, the model treats that class as the positive class and the other class as the negative class. Let $F_{1,f}$ be the F_1 score computed when class "Failure" is the positive one and $F_{1,n}$ be the F_1 score computed when class "No failure" is the positive one. Then, the macro F_1 score is given by Equation (5).

$$MF_1 = \frac{F_{1,f} + F_{1,n}}{2} \tag{5}$$

The macro F_1 score allows for a balanced assessment of the model's performance, as it weighs both the "Failure" and "No failure" classes equally, different than, e.g., accuracy. Hence, macro F_1 is not biased towards the majority class (i.e., "No failure").

The range of RW and PW sizes is adapted to the dynamics of the machines (a slow varying behaviour requires testing fewer window sizes than a fast-changing one). For the wrapping machine dataset, we use nine values for the PW (0.25, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, and 4 h) and six values for the RW (10, 15, 20, 25, 30, and 35 min). For the blood refrigerator, we use four values for the PW (0.5, 1, 1.5, and 2 h) and five values for the RW (10, 15, 20, 25, and 30 min). For the nitrogen generator dataset, we use six values for the PW (0.5, 1, 1.5, 2, 3, and 5 h) and five values for the RW (10, 15, 20, 25, and 30 min). For all the datasets, the minimum PW size is set to the smallest time span necessary to avoid an unrecoverable machine failure so that the operator can intervene.

The validation procedure is also adapted to the characteristics of the different use cases. In the wrapping machine dataset, there are only 13 alarms, which yield ≈500 *failure* RWs in the whole time series. Thus, the number of failure RWs in the test set would be too small to test the performances adequately. Thus, we adopt a training and evaluation procedure based on k-fold cross-validation (with $k = 5$). Each fold is identified by f_k with $k \in [1, 5]$. The procedure also relies on 10 RUS instances r_{ik} for the test set and 10 RUS

instances p_{ik} for the train set, six algorithms a_j, with $j \in \{$RF, LR, SVM, LSTM, ConvLSTM, Transformer$\}$, and the hyperparameter settings for each algorithm (h_{jl}). Each combination of RW and PW sizes is used to compare the models with a procedure consisting of six steps:

1. For each k-fold split, the data are divided into a train set TrS_k, comprising the folds f_m with $m \neq k$, and a test set TeS_k, corresponding to f_k.
2. For each TrS_k and TeS_k, RUS is applied to balance the classes, obtaining 10 RUS instances r_{ik} for the test set and 10 RUS instances p_{ik} for the train set.
3. For each p_{ik}, the six presented algorithms a_j, with their hyperparameters h_{jl}, are trained on the train folds and evaluated on the RUS instance r_{ik} on the test fold, obtaining $j \times l$ macro F_1 scores on the test set, denoted as s_{ikjl}.
4. For each algorithm and hyperparameter setting, the mean of the results is computed as $m_{jl} = \text{mean}_{i,k}(s_{ikjl})$.
5. Then, the maximum macro F_1 score for each algorithm is computed as $b_j = \max_l(m_{jl})$.
6. Finally, the best macro F_1 score is computed as $B_s = \max_j(b_j)$, and the best algorithm as $B_a = \text{argmax}_j(b_j)$.

For the blood refrigerator and nitrogen generator datasets, a higher number of *failure* RWs is available; thus, k-fold cross-validation is unnecessary. Each dataset is first divided into a train, validation, and test set. The model is trained on the training set, and the best combination of hyperparameters is determined as the one with the best macro F_1 score on the validation set using Bayesian Optimization. Finally, the best configuration of each algorithm is evaluated on the test set.

4. Results

This section reports the macro F_1 scores (from now on referred to as F_1 scores) and the macro average precision and recall of the compared algorithms computed with the abovementioned procedure. It discusses the effect of RW and PW selection on SVM, RF, LR, LSTM, ConvLSTM, and Transformer prediction performances.

4.1. Wrapping Machine

In the wrapping machine case, the cardinality of the minority class samples is significantly influenced by both the PW and the RW, as reading windows spanning multiple sessions are neglected. Figure 6 shows the number of RWs with class “Failure” (i.e., the support) as RW and PW sizes vary. Note that a longer PW is more likely to contain a failure, resulting in a higher probability of the RW being classified as “Failure". Consequently, the number of RWs classified as “Failure” increases for larger PWs (i.e., the support increases). The support also increases with the decrease in RW size because more RWs are paired to a PW of class “Failure”.

Figure 7 shows the best algorithms for each reading and prediction window size (i.e., the B_a values) and their F_1 scores (i.e., the B_s values), and Table 7 compares the F_1 scores of each algorithm as PW and RW vary (i.e., the b_j values). In both cases, lighter backgrounds indicate better performances. Missing results correspond to cases in which at least one fold does not contain RWs of class "Failure”.

Table 7, Figures 7 and 8 show that the F_1 score, on average, increases as the RW size increases until a peak at 20 min and decreases afterwards. Smaller train sets and the lesser relevance of past information can explain this behaviour [85,86]. For example, considering a PW of 15 min, the support increases of a factor ≈4 from an RW of 30 min to an RW of 10 min. The work in [86] reports that decreasing the number of train set samples (in this case, the number of RWs in the train set) can reduce performances and that the best results are achieved for the greatest number of training samples. The work in [85] finds that increasing the amount of historical data (in this case, the RW length) does not necessarily lead to better performance, as irrelevant information may be provided to the predictor. Similarly to [12], the performances tend to decrease as the PW size increases, with the best

performances observed for the smallest PW size. This pattern depends on the fact that the predictions for smaller PWs exploit more recent data than for larger PWs.

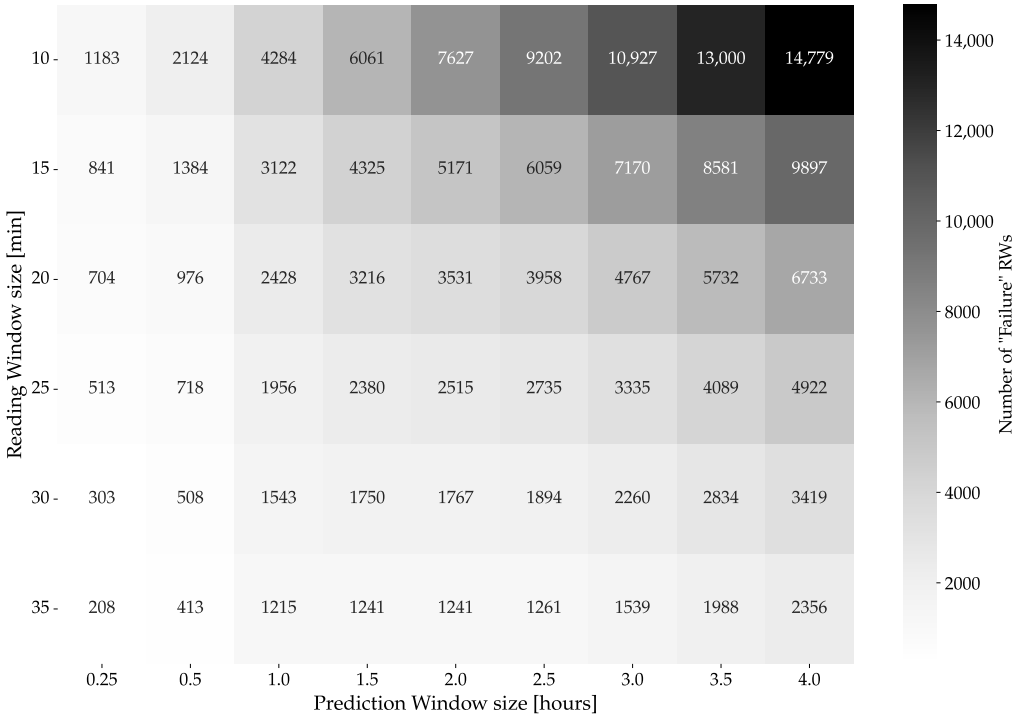


Figure 6. The number of RWs with class "Failure" as RW and PW sizes vary in the wrapping machines dataset. The highest number of "Failure" labels is observed for short RWs and long PWs. Darker colours indicate a higher number of "Failure" RWs.

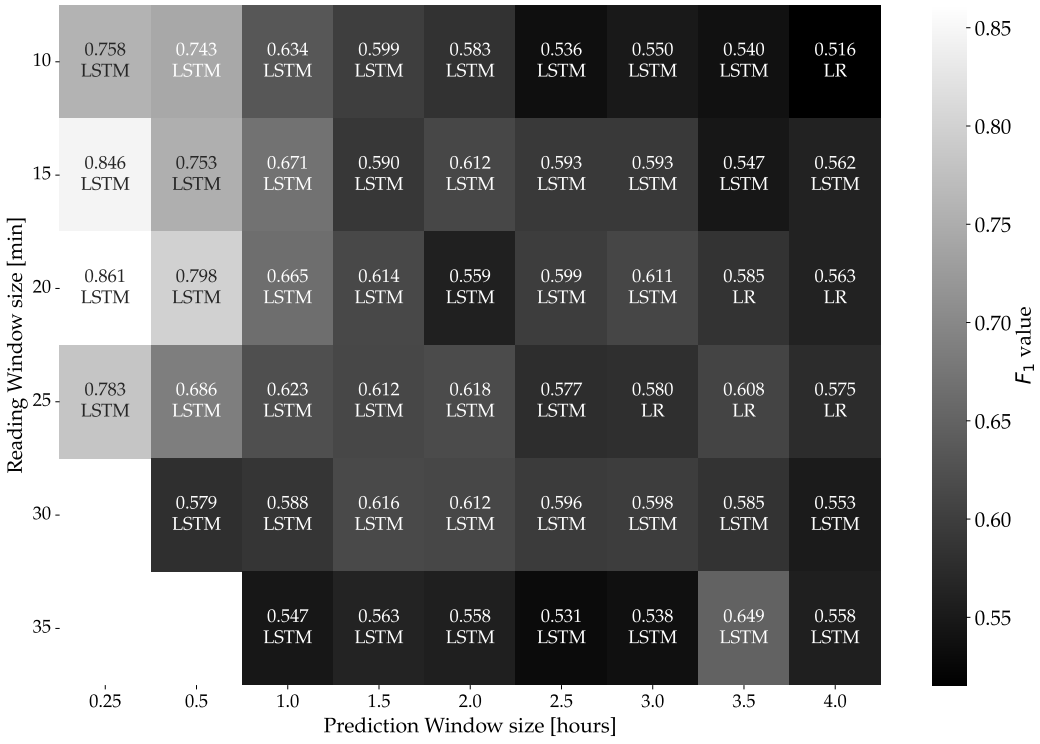


Figure 7. Comparison of the best F_1 scores for each combination of RW and PW sizes for the wrapping machine case study. The heatmap shows each combination's best method and the corresponding F_1 score. Lighter background colours correspond to better results.

Table 7. Classification results on the F_1 metrics for the wrapping machine case study, varying both the reading and prediction window, for SVM, RF, LR, LSTM, ConvLSTM, and Transformers. A lighter background corresponds to better results. Empty cells indicate that the results cannot be computed due to the lack of "Failure" labels in some folds.

Reading Window (RW)	Algorithm	Prediction Window (PW)								
		0.25 h	0.5 h	1.0 h	1.5 h	2.0 h	2.5 h	3.0 h	3.5 h	4.0 h
10 min	ConvLSTM	0.684	0.691	0.577	0.554	0.533	0.517	0.524	0.506	0.483
	LR	0.604	0.598	0.549	0.518	0.503	0.498	0.532	0.514	0.516
	LSTM	0.758	0.743	0.634	0.599	0.583	0.536	0.550	0.540	0.512
	RF	0.577	0.571	0.429	0.417	0.402	0.388	0.382	0.381	0.391
	SVM	0.697	0.654	0.590	0.543	0.521	0.517	0.520	0.485	0.476
	Transformer	0.714	0.646	0.585	0.584	0.540	0.486	0.498	0.482	0.452
15 min	ConvLSTM	0.787	0.683	0.554	0.530	0.547	0.532	0.558	0.514	0.498
	LR	0.742	0.548	0.516	0.530	0.513	0.483	0.526	0.539	0.540
	LSTM	0.846	0.753	0.671	0.590	0.612	0.593	0.593	0.547	0.562
	RF	0.565	0.391	0.410	0.363	0.361	0.376	0.365	0.349	0.367
	SVM	0.753	0.659	0.532	0.524	0.513	0.518	0.505	0.471	0.477
	Transformer	0.818	0.677	0.613	0.567	0.563	0.506	0.513	0.503	0.477
20 min	ConvLSTM	0.807	0.699	0.567	0.559	0.531	0.546	0.551	0.537	0.489
	LR	0.789	0.696	0.587	0.554	0.517	0.518	0.555	0.585	0.563
	LSTM	0.861	0.798	0.665	0.614	0.559	0.599	0.611	0.557	0.557
	RF	0.538	0.448	0.404	0.393	0.381	0.389	0.379	0.417	0.384
	SVM	0.783	0.568	0.499	0.468	0.486	0.485	0.481	0.443	0.468
	Transformer	0.799	0.649	0.584	0.559	0.543	0.513	0.516	0.478	0.438
25 min	ConvLSTM	0.744	0.635	0.527	0.541	0.512	0.557	0.523	0.561	0.524
	LR	0.735	0.620	0.552	0.531	0.519	0.530	0.580	0.608	0.575
	LSTM	0.783	0.686	0.623	0.612	0.618	0.577	0.579	0.568	0.538
	RF	0.641	0.433	0.395	0.379	0.401	0.433	0.410	0.401	0.378
	SVM	0.613	0.512	0.436	0.411	0.416	0.436	0.482	0.413	0.435
	Transformer	0.772	0.660	0.517	0.529	0.504	0.493	0.492	0.449	0.450
30 min	ConvLSTM		0.426	0.490	0.464	0.437	0.470	0.569	0.569	0.528
	LR		0.456	0.510	0.474	0.471	0.495	0.542	0.574	0.532
	LSTM		0.579	0.588	0.616	0.612	0.596	0.598	0.585	0.553
	RF		0.367	0.360	0.353	0.353	0.352	0.349	0.353	0.347
	SVM		0.332	0.348	0.339	0.338	0.343	0.438	0.406	0.405
	Transformer		0.473	0.505	0.489	0.474	0.464	0.443	0.475	0.443
35 min	ConvLSTM			0.396	0.405	0.409	0.419	0.461	0.597	0.492
	LR			0.410	0.393	0.393	0.395	0.473	0.526	0.492
	LSTM			0.547	0.563	0.558	0.531	0.538	0.649	0.558
	RF			0.333	0.333	0.333	0.333	0.332	0.351	0.343
	SVM			0.328	0.328	0.328	0.329	0.375	0.428	0.338
	Transformer			0.516	0.471	0.490	0.464	0.433	0.492	0.464

LSTM performs better than the other algorithms (on average, 61.4% F_1). ConvLSTM is, on average, the second-best algorithm (on average, 54.5% F_1), followed by LR (on average, 54.0% F_1), Transformers (on average, 52.9% F_1), SVM (on average, 47.4% F_1), and RF (on average, 39.8% F_1). Related studies on time series forecasting also show the effectiveness of LSTM over simpler non-DL approaches [80,87,88]. Figure 9 shows that the improvement in performance introduced by LSTM (the best DL algorithm) with respect to LR (the best ML algorithm) is, on average, higher for smaller PWs and lower for larger PWs. The work in [89] also observes that LSTM forecasting performances decrease for a longer prediction horizon. This pattern can be explained by the decrease in the relevance of past information for predicting behaviours far in the future. In addition, ref. [83] observes that the performance improvement between different models tends to decrease as the forecasting horizon enlarges, suggesting that more complex approaches are as effective as

less complex ones when the historical data are far in the past. Overall, our findings show that in the case study, LSTM can capture temporal dependencies effectively and behave better for small PWs (i.e., the observed temporal patterns have a noticeable influence on the predicted class). As observed in [90], LR can exploit time-independent patterns since it does not consider any temporal dependency. This explains why the difference between LSTM and LR performances decreases as the PW increases.

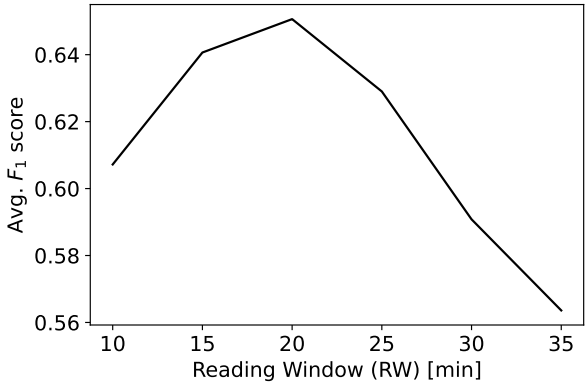


Figure 8. The F_1 score varies depending on the RW and, on average, reaches its peak for an RW of 20 min. This pattern is particularly noticeable for a PW of 0.25 h.

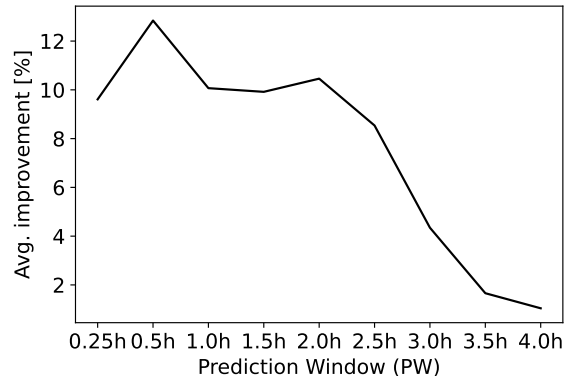


Figure 9. The average improvement of LSTM with respect to LR shows that LSTM benefits more from short PWs since it can capture relevant time-dependent patterns. Note that the values for the two smallest PWs (0.25 and 0.5 h) are not entirely comparable with the others because they do not consider the longest RWs, for which the prediction could not be computed.

Considering LSTM-based networks, the choice of the loss function also has an impact, as summarized in Table 8. As the RW and PW vary, the difficulty of predicting failures changes. In particular, predicting errors becomes harder when the PW increases, especially when less historical information is available (i.e., for small RWs and large PWs). In such cases, the F_1 loss is ineffective, as it maximizes the F_1 score for the "Failure" class. However, a relatively high score (≈66.7%) corresponds to the trivial case in which only the "Failure" class is predicted, and it becomes more challenging to surpass this result in the most difficult instances of the prediction problem. Instead, the BCE loss does not privilege one of the two classes and is more suitable for the more challenging cases, leading to better macro F_1 scores.

Focusing on the ML approaches, RF and SVM show poorer performances. SVM, similarly to LR, is not able to capture time dependencies. However, LR is expected to perform better than SVM, especially when the number of features exceeds the number of samples, as SVM with a Radial Basis Function (RBF) kernel is prone to overfitting [91], whereas LR is a simpler model. In this case, the number of features for both algorithms is given by the number of samples in the reading windows multiplied by the number of time

series variables. On average, there are more features than training windows (≈1.13 features per training sample), which justifies the better performances of LR over SVM (≈1.14 times better in terms of F_1 score, on average).

Table 8. Comparison of the F_1 (in white) and BCE (in gray) loss functions for LSTM. F_1 loss becomes less effective as the classification problem becomes more difficult (i.e., when the PW increases and the RW decreases).

Reading Window (RW)	Prediction Window (PW)								
	0.25 h	**0.5 h**	**1.0 h**	**1.5 h**	**2.0 h**	**2.5 h**	**3.0 h**	**3.5 h**	**4.0 h**
10 min	F_1	F_1	BCE	BCE	BCE	BCE	BCE	BCE	BCE
15 min	F_1	F_1	F_1	BCE	BCE	BCE	BCE	BCE	BCE
20 min	F_1	F_1	F_1	F_1	BCE	BCE	BCE	BCE	BCE
25 min	F_1	F_1	F_1	F_1	F_1	F_1	BCE	BCE	BCE
30 min		F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1
35 min			F_1	F_1	F_1	F_1	F_1	F_1	F_1

RF is the worst approach because each decision tree considers a limited number of features selected randomly, neglecting temporal dependencies, and the number of trees is much smaller than the number of input features (≈2% to ≈13%). For these reasons, the random forest ensembles have weak estimators that cannot capture complex patterns.Increasing the number of estimators or the maximum number of features does not necessarily yield better results. For the combination of PW and RW leading to the best result for RF (i.e., 0.25 h and 25 min, respectively), the best number of trees is 150, and the best number of maximum features is 33% of all the features. Increasing the number of trees to 200 yields a −0.01% variation in the macro F_1 score, and increasing the maximum number of features to 100% yields a −0.1% variation in the macro F_1 score. On average, across all the RW-PW combinations, the best ML algorithm has an F_1 of ≈54.8%, whereas the best DL algorithm (LSTM) has an F_1 of ≈61.4%. DL algorithms are the best choice for the specified fault prediction task in this dataset.

In summary, the best results are obtained using a PW of 0.25 h and an RW of 20 min, which is considered suitable in the industrial setting of the case study. A moderate amount of historical data is sufficient to predict an alert enough in advance to allow the operator to intervene. For these RW and PW values, SVM, LR, LSTM, and ConvLSTM reach their highest F_1 scores (respectively, 0.783, 0.789, 0.861, and 0.807). However, LSTM obtains the best F_1 score, which is ≈7% better than LR and SVM, ≈5% better than ConvLSTM, and ≈32% better than RF. The difference between LSTM and the other approaches is justified by considering that it can consider temporal dependencies explicitly, and benefits are especially significant when recent historical data are used for predicting. Regarding the other DL algorithms, Transformers focus on non-sequential patterns and ConvLSTM focuses on spatial patterns, which makes them less effective for the specified time-series fault prediction task.

4.2. Blood Refrigerator

Table 9 shows the F_1 scores of each algorithm as PW and RW vary, similarly to Table 7. DL algorithms outperform ML algorithms in fewer cases in the blood refrigerator dataset than in the wrapping machine dataset. Such difference can be explained by considering the different nature of the datasets. While the wrapping machine faults are not associated with clearly identifiable short patterns and are caused by a longer wearing of the machine components, in the blood refrigerator, the high product temperature is often observed as a consequence of some short, easily identifiable patterns, exemplified in Figure 4. Such patterns can be found by simpler algorithms (i.e., ML algorithms), and more complex architectures do not lead to significant benefits.

Table 9. Classification results on the F_1 metrics for the blood refrigerator case study, varying both the reading and prediction window, for SVM, RF, LR, LSTM, Transformer, and ConvLSTM. A lighter background corresponds to better results.

Reading Window (RW)	Algorithm	Prediction Window (PW)			
		0.5 h	1.0 h	1.5 h	2.0 h
10 min	ConvLSTM	0.750	0.612	0.634	0.575
	LR	0.729	0.602	0.610	0.602
	LSTM	0.735	0.615	0.633	0.633
	RF	0.765	0.568	0.524	0.546
	SVM	0.748	0.635	0.634	0.612
	Transformer	0.794	0.621	0.558	0.659
15 min	ConvLSTM	0.727	0.615	0.633	0.615
	LR	0.727	0.599	0.608	0.608
	LSTM	0.782	0.657	0.599	0.563
	RF	0.773	0.624	0.569	0.548
	SVM	0.742	0.650	0.640	0.609
	Transformer	0.787	0.551	0.556	0.603
20 min	ConvLSTM	0.680	0.624	0.617	0.576
	LR	0.712	0.605	0.598	0.608
	LSTM	0.760	0.643	0.607	0.568
	RF	0.814	0.619	0.574	0.550
	SVM	0.744	0.648	0.636	0.614
	Transformer	0.795	0.633	0.670	0.659
25 min	ConvLSTM	0.725	0.634	0.627	0.542
	LR	0.721	0.605	0.599	0.599
	LSTM	0.779	0.665	0.520	0.489
	RF	0.822	0.599	0.578	0.586
	SVM	0.738	0.651	0.634	0.613
	Transformer	0.758	0.658	0.607	0.670
30 min	ConvLSTM	0.730	0.544	0.601	0.588
	LR	0.724	0.598	0.603	0.587
	LSTM	0.804	0.633	0.577	0.477
	RF	0.835	0.606	0.571	0.555
	SVM	0.742	0.635	0.632	0.606
	Transformer	0.728	0.646	0.613	0.519

Figure 10 presents the algorithm with the best F_1 score for each RW and PW pair. The results highlight the small variation of performances across the algorithms. On average, the best ML algorithm for each RW-PW combination has an F_1 of ≈67.3%, whereas the best DL algorithm for each RW-PW combination has an F_1 of ≈67.8%.

In the case of the blood refrigerator, the absence of sessions leads to negligible differences in the support and does not affect results significantly. In this case, the average F_1 score for different RWs varies between ≈67.4% and ≈69.8%, a small variation (≈2.4%) compared to the one of the wrapping machine dataset (≈9%). This result also suggests that the amount of historical data has a negligible effect on this dataset and that short RWs already lead to high F_1 scores.

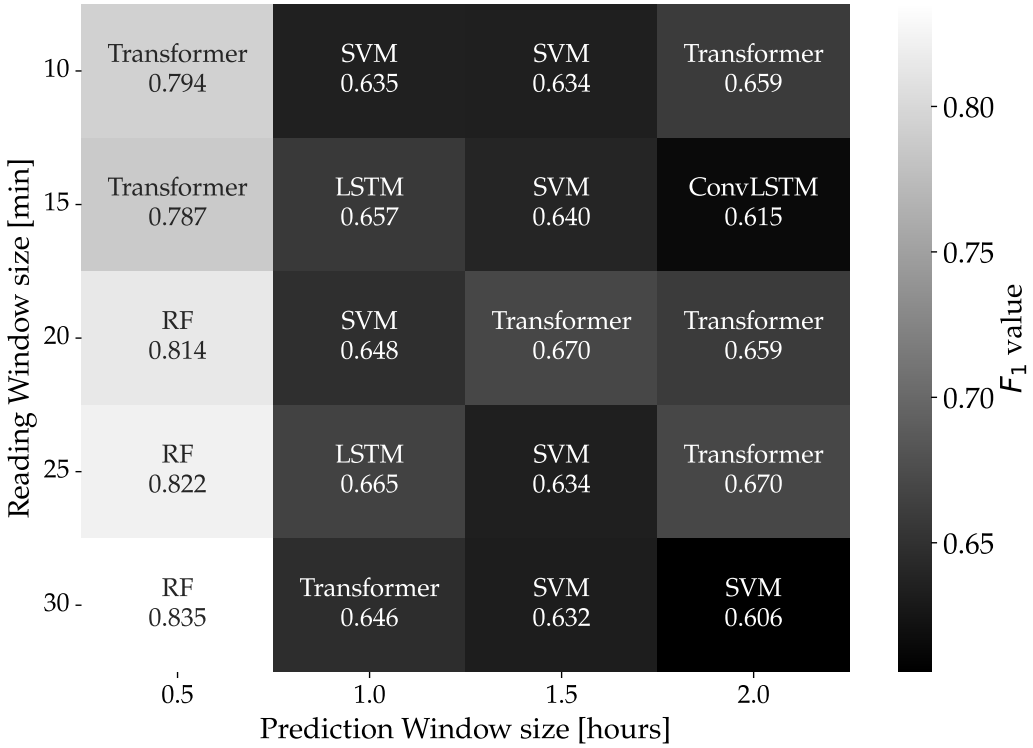


Figure 10. Comparison of the best F_1 scores for each combination of RW and PW sizes for the blood refrigerator case study. The heatmap shows each combination's best method and the corresponding F_1 score. Lighter background colours correspond to better results.

4.3. Nitrogen Generator

Table 10 compares the F_1 scores of each algorithm as PW and RW vary. Also in this case, the performances of ML and DL algorithms are similar, suggesting that as the dataset becomes simpler, more complex algorithms become less effective and often worse than simple ones. In particular, the best DL algorithms for each RW-PW combination reach, on average, an F_1 score of 89.1%, whereas the best ML algorithms for each RW-PW combination reach an average of 89.0%. As in the blood refrigerators case, this result can be explained considering the easily identifiable and similar anomalous patterns preceding faults, as visible in Figure 5.

Figure 11 presents the algorithm with the best F_1 score for each RW and PW pair. The results highlight the similar performances of different algorithms on all the RWs and PWs. In particular, the average F_1 score of the best algorithm varies between 87.9% and 88.5% as the RW changes, a smaller difference (≈0.6%) compared to the other datasets.

In this case, RF is, on average, the best algorithm (with an F_1 score of ≈87.3%). It benefits from simple and repetitive patterns in the data that it can capture effectively. Other algorithms have similar performances. Transformers have an average F_1 of ≈86.6%, LSTM has an average F_1 of ≈85.6%, SVM has an average F_1 of ≈85.6%, ConvLSTM has an average F_1 of ≈85.3%, and LR has an average F_1 of ≈78.4%.

Table 10. Classification results on the F_1 metrics for the nitrogen generator case study, varying both the reading and prediction window, for SVM, RF, LR, LSTM, Transformer, and ConvLSTM. A lighter background corresponds to better results.

Reading Window (RW)	Algorithm	Prediction Window (PW)					
		0.5 h	1.0 h	1.5 h	2.0 h	3.0 h	5.0 h
10 min	ConvLSTM	0.875	0.900	0.873	0.879	0.810	0.792
	LR	0.812	0.824	0.841	0.835	0.660	0.674
	LSTM	0.874	0.872	0.869	0.856	0.865	0.806
	RF	0.931	0.885	0.883	0.883	0.844	0.838
	SVM	0.903	0.882	0.885	0.871	0.807	0.771
	Transformer	0.874	0.879	0.871	0.890	0.828	0.811
15 min	ConvLSTM	0.830	0.904	0.798	0.868	0.830	0.848
	LR	0.837	0.837	0.840	0.828	0.663	0.676
	LSTM	0.818	0.892	0.887	0.879	0.830	0.807
	RF	0.919	0.892	0.876	0.878	0.834	0.838
	SVM	0.894	0.879	0.875	0.874	0.810	0.784
	Transformer	0.892	0.888	0.893	0.883	0.839	0.821
20 min	ConvLSTM	0.888	0.893	0.882	0.876	0.856	0.812
	LR	0.832	0.837	0.841	0.826	0.670	0.678
	LSTM	0.834	0.885	0.906	0.878	0.813	0.814
	RF	0.902	0.890	0.873	0.882	0.843	0.834
	SVM	0.882	0.877	0.873	0.881	0.815	0.795
	Transformer	0.896	0.891	0.896	0.888	0.813	0.825
25 min	ConvLSTM	0.842	0.872	0.887	0.890	0.809	0.823
	LR	0.845	0.838	0.854	0.827	0.675	0.680
	LSTM	0.814	0.893	0.888	0.879	0.832	0.811
	RF	0.914	0.883	0.873	0.880	0.857	0.836
	SVM	0.872	0.882	0.873	0.887	0.829	0.803
	Transformer	0.877	0.861	0.905	0.898	0.838	0.836
30 min	ConvLSTM	0.752	0.881	0.887	0.882	0.850	0.796
	LR	0.863	0.848	0.864	0.825	0.691	0.687
	LSTM	0.848	0.865	0.910	0.885	0.852	0.812
	RF	0.892	0.880	0.873	0.871	0.868	0.839
	SVM	0.863	0.878	0.879	0.889	0.837	0.817
	Transformer	0.868	0.879	0.912	0.898	0.821	0.820

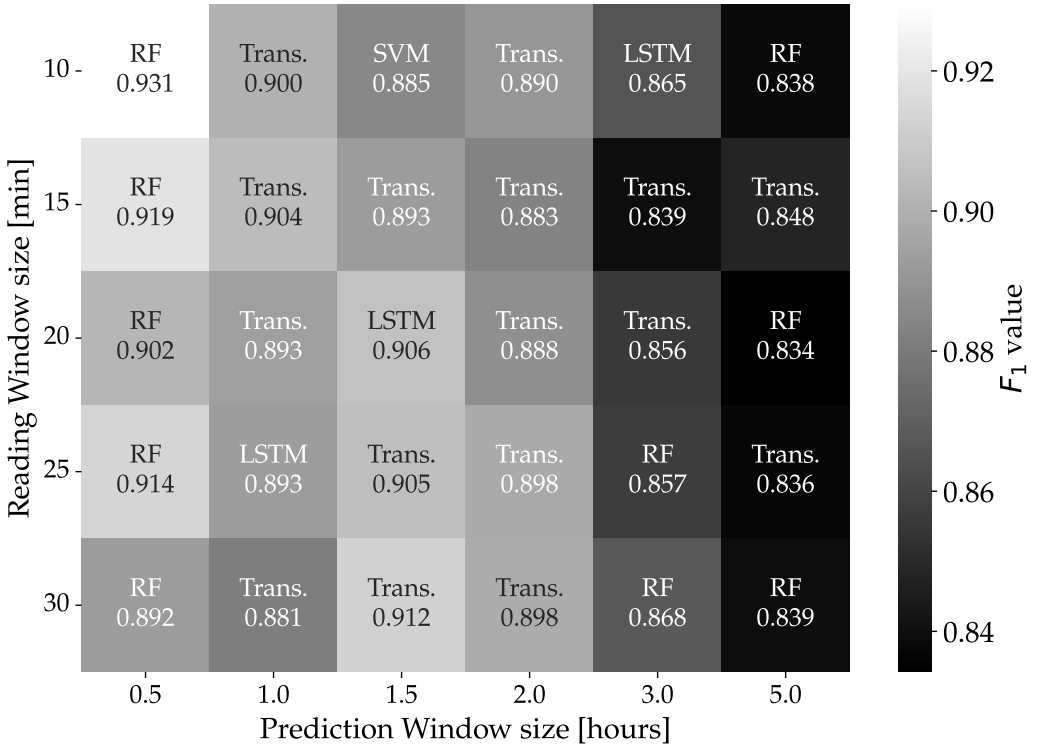


Figure 11. Comparison of the best F_1 scores for each combination of RW and PW sizes for the nitrogen generator case study. The heatmap shows each combination's best method and the corresponding F_1 score, where "Trans." indicates the Transformer. Lighter background colours correspond to better results.

4.4. Execution Time

Table 11 summarizes the training times when using a GeForce RTX 2080 Ti GPU for DL algorithms and an Intel Core i9-9900K CPU for ML algorithms. Times are expressed in minutes and refer to the average time for training for the best RW-PW combination with a single set of hyperparameters. The training time is significantly higher for DL algorithms (up to 115×), making ML algorithms a better choice than DL algorithms when their macro F_1 score is similar (i.e., in the blood refrigerator and the nitrogen generator cases).

Table 11. Comparison of the presented algorithms training times, in minutes.

	Wrapping Machine	**Blood Refrigerator**	**Nitrogen Generator**
RF	1.40	0.65	0.04
SVM	2.10	0.72	0.15
LR	1.90	0.68	0.10
LSTM	20.00	4.00	4.80
ConvLSTM	46.00	1.50	1.93
Transformer	19.00	1.70	3.85

5. Conclusions

Failure prediction on industrial multivariate data is crucial for implementing effective predictive maintenance strategies to reduce downtime and increase productivity and operational time. However, achieving this goal with non-neural machine learning and deep learning models is challenging and requires a deep understanding of the input data.

The illustrated case studies show the importance of setting meaningful prediction and reading windows consistent with domain-specific requirements. Experimental results demonstrate that basic, general-purpose algorithms, such as logistic regression, already

achieve acceptable performances on complex datasets, where complexity is the mean of spectral entropies, as defined in Section 3.2.

In the wrapping machine case study, logistic regression attains a macro F_1 score of 0.789 with a prediction window of 15 min and a reading window of 20 min. In the same case study, LR is outperformed by LSTM, a complex model that exploits temporal dependencies, with a margin of more than 7% on average on all RW-PW pairs. In the best-case scenario of a prediction window of 15 min and a reading window of 20 min, the LSTM model achieves a notable macro F_1 score of 0.861. However, the performance gap declines as the prediction horizon enlarges because the relevance of the temporal dependencies between the RW and the PW fades out. Ultimately, an LSTM model is maximally effective for short PWs when the influence of the reading window historical data is expected to be more critical and becomes less effective as the prediction window increases.

However, the better performances obtained with DL algorithms are negligible in simpler datasets, where easily identifiable repetitive patterns can be found. The blood refrigerator dataset is simpler, with a few relevant anomalous patterns leading to failures. In that case, ML algorithms perform similarly to DL algorithms, often surpassing the latter. A similar result is obtained for the nitrogen generator dataset. In this case, random forest, which has the worst performance on the wrapping machine dataset, can effectively find simple rule-based, repetitive patterns anticipating failures, achieving the best performances.

The results presented in this paper are valid for the industrial scenario and datasets employed in the experiments. Larger datasets would likely contain more occurrences of multiple alert codes, enabling the study of different types of anomalies.

Our future work will focus on fine-tuning the LSTM model and exploring different architectures (e.g., CNN-based models [92] and Gaussian Processes [93]), hybrid models that combine multiple machine learning and deep learning techniques [94], and digital twin-based approaches that exploit simulations of physical machines. The latter have shown good performance in diverse fields, including aeronautics [95,96], hydraulic systems [97], and facility management [98]. Such a comparison can help identify the most effective approach for different industrial scenarios and enable a better understanding of how different network structures and learning algorithms impact failure prediction accuracy. Another fundamental research direction for applying LSTM as predictors is the interpretability of their output [99]. As DL models are often used as black boxes, understanding the reasons behind their predictions is crucial in industrial settings where the model's output is used to take preventive maintenance action.

Author Contributions: Conceptualization, N.O.P.V., F.F., and P.F.; methodology, N.O.P.V., F.F., and P.F.; software, N.O.P.V., F.F., and P.F.; validation, N.O.P.V., F.F., and P.F.; formal analysis, N.O.P.V., F.F., and P.F.; investigation, N.O.P.V., F.F., and P.F.; resources, N.O.P.V., F.F., and P.F.; data curation, N.O.P.V., F.F., and P.F.; writing—original draft preparation, N.O.P.V., F.F., and P.F.; writing—review and editing, N.O.P.V., F.F., and P.F.; visualization, N.O.P.V., F.F., and P.F.; supervision, P.F.; project administration, P.F.; funding acquisition, P.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union's Horizon 2020 project PRECEPT, grant number 958284.

Data Availability Statement: Datasets and code have been made available at https://github.com/nicolopinci/polimi_failure_prediction, accessed on 19 May 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bousdekis, A.; Apostolou, D.; Mentzas, G. Predictive Maintenance in the 4th Industrial Revolution: Benefits, Business Opportunities, and Managerial Implications. *IEEE Eng. Manag. Rev.* **2020**, *48*, 57–62. [CrossRef]
2. Zonta, T.; da Costa, C.A.; da Rosa Righi, R.; de Lima, M.J.; da Trindade, E.S.; Li, G.P. Predictive maintenance in the Industry 4.0: A systematic literature review. *Comput. Ind. Eng.* **2020**, *150*, 106889. [CrossRef]

3. Dalzochio, J.; Kunst, R.; Pignaton, E.; Binotto, A.; Sanyal, S.; Favilla, J.; Barbosa, J. Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges. *Comput. Ind.* **2020**, *123*, 103298. [CrossRef]
4. Sheut, C.; Krajewski, L.J. A decision model for corrective maintenance management. *Int. J. Prod. Res.* **1994**, *32*, 1365–1382. [CrossRef]
5. Wang, Y.; Deng, C.; Wu, J.; Wang, Y.; Xiong, Y. A corrective maintenance scheme for engineering equipment. *Eng. Fail. Anal.* **2014**, *36*, 269–283. [CrossRef]
6. Meller, R.D.; Kim, D.S. The impact of preventive maintenance on system cost and buffer size. *Eur. J. Oper. Res.* **1996**, *95*, 577–591. [CrossRef]
7. Wu, S.; Zuo, M. Linear and Nonlinear Preventive Maintenance Models. *IEEE Trans. Reliab.* **2010**, *59*, 242–249. [CrossRef]
8. Liang, H.; Song, L.; Wang, J.; Guo, L.; Li, X.; Liang, J. Robust unsupervised anomaly detection via multi-time scale DCGANs with forgetting mechanism for industrial multivariate time series. *Neurocomputing* **2021**, *423*, 444–462. [CrossRef]
9. Tian, Z.; Zhuo, M.; Liu, L.; Chen, J.; Zhou, S. Anomaly detection using spatial and temporal information in multivariate time series. *Sci. Rep.* **2023**, *13*, 4400. [CrossRef]
10. Salfner, F.; Lenk, M.; Malek, M. A survey of online failure prediction methods. *ACM Comput. Surv.* **2010**, *42*, 1–42. [CrossRef]
11. García, F.P.; Pedregal, D.J.; Roberts, C. Time series methods applied to failure prediction and detection. *Reliab. Eng. Syst. Saf.* **2010**, *95*, 698–703. [CrossRef]
12. 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.* **2022**, *40*, 1449–1462. [CrossRef]
13. Box, G.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Wiley: Hoboken, NJ, USA, 2016.
14. Łuczak, D.; Brock, S.; Siembab, K. Cloud Based Fault Diagnosis by Convolutional Neural Network as Time–Frequency RGB Image Recognition of Industrial Machine Vibration with Internet of Things Connectivity. *Sensors* **2023**, *23*, 3755. [CrossRef] [PubMed]
15. Pertselakis, M.; Lampathaki, F.; Petrali, P. Predictive Maintenance in a Digital Factory Shop-Floor: Data Mining on Historical and Operational Data Coming from Manufacturers' Information Systems. In *Lecture Notes in Business Information Processing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 120–131. [CrossRef]
16. Khorsheed, R.M.; Beyca, O.F. An integrated machine learning: Utility theory framework for real-time predictive maintenance in pumping systems. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2020**, *235*, 887–901. [CrossRef]
17. 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. [CrossRef]
18. Kaparthi, S.; Bumblauskas, D. Designing predictive maintenance systems using decision tree-based machine learning techniques. *Int. J. Qual. Reliab. Manag.* **2020**, *37*, 659–686. [CrossRef]
19. Alves, F.; Badikyan, H.; Moreira, H.A.; Azevedo, J.; Moreira, P.M.; Romero, L.; Leitao, P. Deployment of a Smart and Predictive Maintenance System in an Industrial Case Study. In Proceedings of the 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, The Netherlands, 17–19 June 2020. [CrossRef]
20. Dix, M.; Chouhan, A.; Sinha, M.; Singh, A.; Bhattarai, S.; Narkhede, S.; Prabhune, A. An AI-based Alarm Prediction in Industrial Process Control Systems. In Proceedings of the 2022 IEEE International Conference on Big Data and Smart Computing (BigComp), Daegu, Republic of Korea, 17–20 January 2022. [CrossRef]
21. Colone, L.; Dimitrov, N.; Straub, D. Predictive repair scheduling of wind turbine drive-train components based on machine learning. *Wind Energy* **2019**, *22*, 17–19. [CrossRef]
22. Leahy, K.; Gallagher, C.; O'Donovan, P.; Bruton, K.; O'Sullivan, D. A Robust Prescriptive Framework and Performance Metric for Diagnosing and Predicting Wind Turbine Faults Based on SCADA and Alarms Data with Case Study. *Energies* **2018**, *11*, 1738. [CrossRef]
23. Bonnevay, S.; Cugliari, J.; Granger, V. Predictive Maintenance from Event Logs Using Wavelet-Based Features: An Industrial Application. In *Advances in Intelligent Systems and Computing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 132–141. [CrossRef]
24. Barraza, J.F.; Bräuning, L.G.; Perez, R.B.; Morais, C.B.; Martins, M.R.; Droguett, E.L. Deep learning health state prognostics of physical assets in the Oil and Gas industry. *Proc. Inst. Mech. Eng. Part O: J. Risk Reliab.* **2020**, *236*, 598–616. [CrossRef]
25. Kusiak, A.; Verma, A. A Data-Mining Approach to Monitoring Wind Turbines. *IEEE Trans. Sustain. Energy* **2012**, *3*, 150–157. [CrossRef]
26. Li, H.; Parikh, D.; He, Q.; Qian, B.; Li, Z.; Fang, D.; Hampapur, A. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transp. Res. Part C Emerg. Technol.* **2014**, *45*, 17–26. [CrossRef]
27. Forbicini, F.; Pinciroli Vago, N.O.; Fraternali, P. Time Series Analysis in Compressor-Based Machines: A Survey. *arXiv* **2024**, arXiv:2402.17802. [CrossRef]
28. Laptev, N.; Amizadeh, S.; Flint, I. Generic and Scalable Framework for Automated Time-series Anomaly Detection. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015. [CrossRef]

29. Ren, H.; Xu, B.; Wang, Y.; Yi, C.; Huang, C.; Kou, X.; Xing, T.; Yang, M.; Tong, J.; Zhang, Q. Time-Series Anomaly Detection Service at Microsoft. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Sydney, Australia, 10–13 August 2015. [CrossRef]
30. Chen, K.; Pashami, S.; Fan, Y.; Nowaczyk, S. Predicting Air Compressor Failures Using Long Short Term Memory Networks. In *Progress in Artificial Intelligence*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 596–609. [CrossRef]
31. Leukel, J.; González, J.; Riekert, M. Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *J. Manuf. Syst.* **2021**, *61*, 87–96. [CrossRef]
32. Zangrando, N.; Fraternali, P.; Petri, M.; Pinciroli Vago, N.O.; González, S.L.H. Anomaly detection in quasi-periodic energy consumption data series: A comparison of algorithms. *Energy Inform.* **2022**, *5*, 62. [CrossRef]
33. Carrera, D.; Manganini, F.; Boracchi, G.; Lanzarone, E. Defect Detection in SEM Images of Nanofibrous Materials. *IEEE Trans. Ind. Inform.* **2017**, *13*, 551–561. [CrossRef]
34. Si, W.; Yang, Q.; Wu, X. Material Degradation Modeling and Failure Prediction Using Microstructure Images. *Technometrics* **2018**, *61*, 246–258. [CrossRef]
35. Bionda, A.; Frittoli, L.; Boracchi, G. Deep Autoencoders for Anomaly Detection in Textured Images Using CW-SSIM. In *Image Analysis and Processing – ICIAP 2022*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 669–680. [CrossRef]
36. Xue, Z.; Dong, X.; Ma, S.; Dong, W. A Survey on Failure Prediction of Large-Scale Server Clusters. In Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), Qingdao, China, 30 July–1 August 2007. [CrossRef]
37. Ramezani, S.B.; Killen, B.; Cummins, L.; Rahimi, S.; Amirlatifi, A.; Seale, M. A Survey of HMM-based Algorithms in Machinery Fault Prediction. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021. [CrossRef]
38. Yoon, A.S.; Lee, T.; Lim, Y.; Jung, D.; Kang, P.; Kim, D.; Park, K.; Choi, Y. Semi-supervised Learning with Deep Generative Models for Asset Failure Prediction. *arXiv* **2017**, arXiv:1709.00845. [CrossRef]
39. Zhao, M.; Furuhata, R.; Agung, M.; Takizawa, H.; Soma, T. Failure Prediction in Datacenters Using Unsupervised Multimodal Anomaly Detection. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020. [CrossRef]
40. Nowaczyk, S.; Prytz, R.; Rögnvaldsson, T.; Byttner, S. Towards a machine learning algorithm for predicting truck compressor failures using logged vehicle data. In Proceedings of the 12th Scandinavian Conference on Artificial Intelligence, Aalborg, Denmark, 20–22 November 2013; IOS Press: Amsterdam, The Netherlands, 2013; pp. 205–214.
41. Prytz, R.; Nowaczyk, S.; Rögnvaldsson, T.; Byttner, S. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Eng. Appl. Artif. Intell.* **2015**, *41*, 139–150. [CrossRef]
42. Canizo, M.; Onieva, E.; Conde, A.; Charramendieta, S.; Trujillo, S. Real-time predictive maintenance for wind turbines using Big Data frameworks. In Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, 19–21 June 2017. [CrossRef]
43. Xiang, S.; Huang, D.; Li, X. A Generalized Predictive Framework for Data Driven Prognostics and Diagnostics using Machine Logs. In Proceedings of the TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Republic of Korea, 28–31 October 2018. [CrossRef]
44. Mishra, K.; Manjhi, S.K. Failure Prediction Model for Predictive Maintenance. In Proceedings of the 2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Bangalore, India, 23–24 November 2018. [CrossRef]
45. Kulkarni, K.; Devi, U.; Sirighee, A.; Hazra, J.; Rao, P. Predictive Maintenance for Supermarket Refrigeration Systems Using Only Case Temperature Data. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018. [CrossRef]
46. Susto, G.A.; Schirru, A.; Pampuri, S.; McLoone, S.; Beghi, A. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Trans. Ind. Inform.* **2015**, *11*, 812–820. [CrossRef]
47. Hamaide, V.; Glineur, F. Predictive Maintenance of a Rotating Condenser Inside a Synchrocyclotron. In Proceedings of the 28th Belgian Dutch Conference on Machine Learning (Benelearn 2019), Brussels, Belgium, 6–8 November 2019.
48. Orrù, P.F.; Zoccheddu, A.; Sassu, L.; Mattia, C.; Cozza, R.; Arena, S. Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry. *Sustainability* **2020**, *12*, 4776. [CrossRef]
49. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
50. Fagerström, J.; Bång, M.; Wilhelms, D.; Chew, M.S. LiSep LSTM: A Machine Learning Algorithm for Early Detection of Septic Shock. *Sci. Rep.* **2019**, *9*, 15132. [CrossRef]
51. Aung, N.N.; Pang, J.; Chua, M.C.H.; Tan, H.X. A novel bidirectional LSTM deep learning approach for COVID-19 forecasting. *Sci. Rep.* **2023**, *13*. [CrossRef] [PubMed]
52. Jin, L.; Wenbo, H.; You, J.; Lei, W.; Fei, J. A ConvLSTM-Based Approach to Wind Turbine Gearbox Condition Prediction. In *Proceedings of the 7th PURPLE MOUNTAIN FORUM on Smart Grid Protection and Control (PMF2022)*; Springer Nature Singapore: Singapore, 2023; pp. 529–545. [CrossRef]

53. Alos, A.; Dahrouj, Z. Using MLSTM and Multioutput Convolutional LSTM Algorithms for Detecting Anomalous Patterns in Streamed Data of Unmanned Aerial Vehicles. *IEEE Aerosp. Electron. Syst. Mag.* **2022**, *37*, 6–15. [CrossRef]
54. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; WOO, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Proceedings of the Advances in Neural Information Processing Systems*; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Scotland, UK, 2015; Volume 28.
55. Szarek, D.; Jabłoński, I.; Zimroz, R.; Wyłomańska, A. Non-Gaussian feature distribution forecasting based on ConvLSTM neural network and its application to robust machine condition prognosis. *Expert Syst. Appl.* **2023**, *230*, 120588. [CrossRef]
56. Wu, X.; Geng, J.; Liu, M.; Song, Z.; Song, H. Prediction of Node Importance of Power System Based on ConvLSTM. *Energies* **2022**, *15*, 3678. [CrossRef]
57. Tuli, S.; Casale, G.; Jennings, N.R. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *arXiv* **2022**, arXiv:2201.07284. [CrossRef]
58. Huang, S.; Liu, Y.; Fung, C.; He, R.; Zhao, Y.; Yang, H.; Luan, Z. HitAnomaly: Hierarchical Transformers for Anomaly Detection in System Log. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2064–2076. [CrossRef]
59. Jin, Y.; Hou, L.; Chen, Y. A Time Series Transformer based method for the rotating machinery fault diagnosis. *Neurocomputing* **2022**, *494*, 379–395. [CrossRef]
60. Wu, B.; Cai, W.; Cheng, F.; Chen, H. Simultaneous-fault diagnosis considering time series with a deep learning transformer architecture for air handling units. *Energy Build.* **2022**, *257*, 111608. [CrossRef]
61. Gao, P.; Guan, L.; Hao, J.; Chen, Q.; Yang, Y.; Qu, Z.; Jin, M. Fault Prediction in Electric Power Communication Network Based on Improved DenseNet. In Proceedings of the 2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Beijing, China, 14–16 June 2023. [CrossRef]
62. Tang, L.; Lv, H.; Yang, F.; Yu, L. Complexity testing techniques for time series data: A comprehensive literature review. *Chaos Solitons Fractals* **2015**, *81*, 117–135. [CrossRef]
63. Inouye, T.; Shinosaki, K.; Sakamoto, H.; Toi, S.; Ukai, S.; Iyama, A.; Katsuda, Y.; Hirano, M. Quantification of EEG irregularity by use of the entropy of the power spectrum. *Electroencephalogr. Clin. Neurophysiol.* **1991**, *79*, 204–210. [CrossRef]
64. Saxena, S.; Odono, V.; Uba, J.; Nelson, J.M.; Lewis, W.L.; Shulman, I.A. The Risk of Bacterial Growth in Units of Blood that Have Warmed to More Than 10 °C. *Am. J. Clin. Pathol.* **1990**, *94*, 80–83. [CrossRef] [PubMed]
65. Blaine, K.P.; Cortés-Puch, I.; Sun, J.; Wang, D.; Solomon, S.B.; Feng, J.; Gladwin, M.T.; Kim-Shapiro, D.B.; Basu, S.; Perlegas, A.; et al. Impact of different standard red blood cell storage temperatures on human and canine RBC hemolysis and chromium survival. *Transfusion* **2018**, *59*, 347–358. [CrossRef]
66. Aalaei, S.; Amini, S.; Keramati, M.R.; Shahraki, H.; Abu-Hanna, A.; Eslami, S. Blood bag temperature monitoring system. In *e-Health–For Continuity of Care*; IOS Press: Amsterdam, The Netherlands, 2014; pp. 730–734.
67. Tanco, M.L.; Tanaka, D.P. Recent Advances on Carbon Molecular Sieve Membranes (CMSMs) and Reactors. *Processes* **2016**, *4*, 29. [CrossRef]
68. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection. *ACM Comput. Surv.* **2009**, *41*, 1–58. [CrossRef]
69. Prusa, J.; Khoshgoftaar, T.M.; Dittman, D.J.; Napolitano, A. Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data. In Proceedings of the 2015 IEEE International Conference on Information Reuse and Integration, San Francisco, CA, USA, 13–15 August 2015. [CrossRef]
70. Zuech, R.; Hancock, J.; Khoshgoftaar, T.M. Detecting web attacks using random undersampling and ensemble learners. *J. Big Data* **2021**, *8*, 75. [CrossRef]
71. Braga, F.C.; Roman, N.T.; Falceta-Gonçalves, D. The Effects of Under and Over Sampling in Exoplanet Transit Identification with Low Signal-to-Noise Ratio Data. In *Lecture Notes in Computer Science*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 107–121. [CrossRef]
72. Hosenie, Z.; Lyon, R.; Stappers, B.; Mootoovaloo, A.; McBride, V. Imbalance learning for variable star classification. *Mon. Not. R. Astron. Soc.* **2020**, *493*, 6050–6059. [CrossRef]
73. Cui, X.; Li, Z.; Hu, Y. Similar seismic moment release process for shallow and deep earthquakes. *Nat. Geosci.* **2023**, *16*, 454–460. [CrossRef]
74. Pereira, P.J.; Pereira, A.; Cortez, P.; Pilastri, A. A Comparison of Machine Learning Methods for Extremely Unbalanced Industrial Quality Data. In *Lecture Notes in Computer Science*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 561–572. [CrossRef]
75. Saripuddin, M.; Suliman, A.; Syarmila Sameon, S.; Jorgensen, B.N. Random Undersampling on Imbalance Time Series Data for Anomaly Detection. In Proceedings of the 2021 The 4th International Conference on Machine Learning and Machine Intelligence (MLMI'21), Hangzhou, China, 17–19 September 2021. [CrossRef]
76. Vuttipittayamongkol, P.; Arreeras, T. Data-driven Industrial Machine Failure Detection in Imbalanced Environments. In Proceedings of the 2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Kuala Lumpur, Malaysia, 7–10 December 2022. [CrossRef]
77. Stefanski, L.A.; Carroll, R.J.; Ruppert, D. Optimally hounded score functions for generalized linear models with applications to logistic regression. *Biometrika* **1986**, *73*, 413–424. [CrossRef]
78. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
79. Sánchez A, V.D. Advanced support vector machines and kernel methods. *Neurocomputing* **2003**, *55*, 5–20. [CrossRef]

80. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The Performance of LSTM and BiLSTM in Forecasting Time Series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019. [CrossRef]
81. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 11121–11128. [CrossRef]
82. Bénédict, G.; Koops, V.; Odijk, D.; de Rijke, M. sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification. *arXiv* **2021**, arXiv:2108.10566. [CrossRef]
83. Abduljabbar, R.L.; Dia, H.; Tsai, P.W. Unidirectional and Bidirectional LSTM Models for Short-Term Traffic Prediction. *J. Adv. Transp.* **2021**, *2021*, 5589075. [CrossRef]
84. Tang, Z.; Wu, B.; Wu, W.; Ma, D. Fault Detection via 2.5D Transformer U-Net with Seismic Data Pre-Processing. *Remote Sens.* **2023**, *15*, 1039. [CrossRef]
85. Zargoush, M.; Sameh, A.; Javadi, M.; Shabani, S.; Ghazalbash, S.; Perri, D. The impact of recency and adequacy of historical information on sepsis predictions using machine learning. *Sci. Rep.* **2021**, *11*, 20869. [CrossRef]
86. Nguyen, Q.H.; Ly, H.B.; Ho, L.S.; Al-Ansari, N.; Le, H.V.; Tran, V.Q.; Prakash, I.; Pham, B.T. Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil. *Math. Probl. Eng.* **2021**, *2021*, 4832864. [CrossRef]
87. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. A Comparison of ARIMA and LSTM in Forecasting Time Series. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018. [CrossRef]
88. Rahimzad, M.; Nia, A.M.; Zolfonoon, H.; Soltani, J.; Mehr, A.D.; Kwon, H.H. Performance Comparison of an LSTM-based Deep Learning Model versus Conventional Machine Learning Algorithms for Streamflow Forecasting. *Water Resour. Manag.* **2021**, *35*, 4167–4187. [CrossRef]
89. Malakar, S.; Goswami, S.; Ganguli, B.; Chakrabarti, A.; Roy, S.S.; Boopathi, K.; Rangaraj, A.G. Designing a long short-term network for short-term forecasting of global horizontal irradiance. *SN Appl. Sci.* **2021**, *3*, 477. [CrossRef]
90. Allam, A.; Nagy, M.; Thoma, G.; Krauthammer, M. Neural networks versus Logistic regression for 30 days all-cause readmission prediction. *Sci. Rep.* **2019**, *9*, 9277. [CrossRef] [PubMed]
91. Han, D.; Chan, L.; Zhu, N. Flood forecasting using support vector machines. *J. Hydroinformatics* **2007**, *9*, 267–276. [CrossRef]
92. Sherly, S.I.; Mathivanan, G. An efficient honey badger based Faster region CNN for chronc heart Failure prediction. *Biomed. Signal Process. Control* **2023**, *79*, 104165. [CrossRef]
93. Lee, W.J.; Sutherland, J.W. Time to Failure Prediction of Rotating Machinery using Dynamic Feature Extraction and Gaussian Process Regression. *Int. J. Adv. Manuf. Technol.* **2023**, *130*, 2939–2955. [CrossRef]
94. 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]
95. Hu, M.; He, Y.; Lin, X.; Lu, Z.; Jiang, Z.; Ma, B. Digital twin model of gas turbine and its application in warning of performance fault. *Chin. J. Aeronaut.* **2023**, *36*, 449–470. [CrossRef]
96. Liu, Z.; Chen, W.; Zhang, C.; Yang, C.; Chu, H. Data Super-Network Fault Prediction Model and Maintenance Strategy for Mechanical Product Based on Digital Twin. *IEEE Access* **2019**, *7*, 177284–177296. [CrossRef]
97. Wang, L.; Liu, Y.; Yin, H.; Sun, W. Fault diagnosis and predictive maintenance for hydraulic system based on digital twin model. *AIP Adv.* **2022**, *12*, 065213. [CrossRef]
98. Hosamo, H.H.; Nielsen, H.K.; Kraniotis, D.; Svennevig, P.R.; Svidt, K. Digital Twin framework for automated fault source detection and prediction for comfort performance evaluation of existing non-residential Norwegian buildings. *Energy Build.* **2023**, *281*, 112732. [CrossRef]
99. Guo, T.; Lin, T.; Antulov-Fantulin, N. Exploring interpretable LSTM neural networks over multi-variable data. In Proceedings of the 36th International Conference on Machine Learning (PMLR), Beach, CA, USA, 9–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; 2019; Volume 97, pp. 2494–2504.