Expert Systems With Applications 222 (2023) 119767

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

LSTM-based failure prediction for railway rolling stock equipment

Luigi De Simone [a,*], Enzo Caputo [a], Marcello Cinque [a], Antonio Galli [a], Vincenzo Moscato [a], Stefano Russo [a], Guido Cesaro [b], Vincenzo Criscuolo [b], Giuseppe Giannini [b]

[a] *Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy*
[b] *Hitachi Rail STS S.p.A., Via Argine 425, 80147 Napoli, Italy*

ARTICLE INFO

Keywords:
Predictive maintenance
Machine learning
Industry 4.0
Railway rolling stock equipment

ABSTRACT

In the railway domain, rolling stock maintenance affects service operation time and efficiency. Minimizing train unavailability is essential for reducing capital loss and operational costs. To this aim, prediction of failures of rolling stock equipment is crucial to proactively trigger proper maintenance activities. Indeed, *predictive maintenance* is a golden example of the digital transformation within Industry 4.0, which affects several engineering processes in the railway domain. Nowadays, it may leverage artificial intelligence and machine learning algorithms to forecast failures and schedule the optimal time for maintenance actions. Generally, rail systems deteriorate gradually over time or fail directly, leading to data that vary extremely slowly. Indeed, ML approaches for predictive maintenance should consider this type of data to accurately predict and forecast failures.

This paper proposes a methodology based on Long Short-Term Memory deep learning algorithms for predictive maintenance of railway rolling stock equipment. The methodology allows us to properly learn long-term dependencies for gradually changing data, and both predicting and forecasting failures of rail equipment. In the framework of an academic-industrial partnership, the methodology is experimented on a train traction converter cooling system, demonstrating its applicability and benefits. The results show that it outperforms state-of-the-art methods, reaching a failure prediction and forecasting accuracy over 99%, with a *false alarm rate* of ∼0.4% and a mean absolute error in the order of 10^{-4}, respectively.

1. Introduction

The manufacturing industry is going through the so-called *Industry 4.0* revolution, in which we are assisting in the massive integration of physical and digital worlds in production environments. Industrial Internet of Things, Big Data, Artificial Intelligence, and 4G and 5G are some of the enabling technologies of this revolution, which are strongly impacting the transport industry. Such technologies permit the collection of a large amount of data gathered from heterogeneous on-board devices and equipment, distributed on train vehicles and railway track sides.

The huge amount of data gathered from (smart) sensors and transmitted to diagnosis systems, on-board or in a control room, can be leveraged through proper techniques to unveil degradation patterns of components and predict failures in useful time for optimal maintenance decisions. Railway transport players usually rely on planned maintenance, which however brings the side effect of taking unnecessary actions, leading to an increase in operating costs. *Condition-based maintenance* (CBM) is a relatively recent extension, which allows estimating the actual conditions of equipment by performing direct measurements; it improves planned maintenance through proactivity, by repairing or replacing degraded devices when defined conditions are met. *Predictive Maintenance* advances CBM using monitoring data and effective predictive techniques to determine the time when a fault is likely to occur. In this way, companies can plan maintenance operations when actually required, with advantages in cost reduction, reduced mean time to failures, repair stop reduction, and overall profit (Peres, Rocha, Leitao, & Barata, 2018; Sezer, Romero, Guedea, Macchi, & Emmanouilidis, 2018). The obvious benefit is that maintenance is done at the right time, that is, before a fault occurs (preventing long downtime) but not unnecessarily too early, ultimately reducing the unavailability of rolling stock as well as of infrastructure equipment.

Machine Learning (ML) algorithms are becoming today compelling tools for maintenance in many heterogeneous domains, as they support predictive techniques. In Rahhal and Abualnadi (2020) the authors use ML to predict the light bulb failure time. In Cachada et al. (2018) ML is used for the early detection of the occurrence of machine failures. ML techniques have also been proposed for rotating machinery failure prediction (Saon, Hiyama, et al., 2010), for the remaining lifetime

* Corresponding author.
E-mail address: luigi.desimone@unina.it (L. De Simone).

https://doi.org/10.1016/j.eswa.2023.119767
Received 8 June 2022; Received in revised form 24 February 2023; Accepted 24 February 2023
Available online 9 March 2023

prediction of hard disks in computing clusters (Basak, Sengupta, & Dubey, 2019; De Santo, Galli, Gravina, Moscato, & Sperli, 2022), and for Remaining Useful Life (RUL) of wind turbines (Nichenametla, Nandipati, & Waghmare, 2017). A semi-supervised technique, obtained by combining supervised and unsupervised Deep Learning algorithms, is proposed in Ellefsen, Bjørlykhaug, Æsøy, Ushakov, and Zhang (2019) to improve the accuracy in the evaluation of turbofan engine's RUL. Thanks to data made available by the NASA Prognostics Center of Excellence (NASA, 2020), ML techniques are also applied to evaluate the degradation of aircraft engines (Aydin & Guldamlasioglu, 2017; Chen et al., 2020). ML handles high-dimensional and multivariate data and helps to extract hidden relationships within data in complex and dynamic environments. Indeed, ML is gaining popularity to improve the operations and reliability also in the railway domain: a comprehensive survey of studies that apply ML in rail track maintenance tasks is provided by Nakhaee, Hiemstra, Stoelinga, and van Noort (2019). However, the performance of ML algorithms strongly depends on the appropriate choice of the ML technique (Williams, Zander, & Armitage, 2006). Generally, rail systems deteriorate gradually over time or fail directly (Bai, Liu, Wang, Sun, & Wang, 2017), leading to data that vary extremely slowly. Indeed, an ML approach for predictive maintenance should consider this type of data to accurately predict and forecast failures.

In the railway domain context, we define the following research questions in the context of failure prediction for rolling stock equipment, i.e., *(R1)* how to properly learn long-term dependencies for gradually changing data?, *(R2)* how to predict with high accuracy failures and their severity levels?, and *(R3)* how to forecast with high accuracy failures?

This paper presents a deep learning-based methodology which answers the research questions depicted above. In particular, the proposed methodology allows, in an ensemble, predicting and forecasting failures to plan maintenance interventions before their occurrence, optimizing the overall cost and duration of maintenance interventions. It is based on Long Short-Term Memory (LSTM) networks, which are an extension of recurrent neural networks (RNN), explicitly designed to learn long-term dependencies for gradually changing data (Hochreiter & Schmidhuber, 1997). The proposed methodology analyzes data from the many sensors connected to the various subsystems composing a railway vehicle. The experimentation focuses on the train traction converter cooling subsystem, one of the most critical on-board systems. We analyzed operational data of a train fleet spanning several months. The framework enables classic statistical data analysis, like trend estimation (e.g., Mann–Kendall) and correlation, to identify coarse-grained relationships between dataset features and failure characterization. Then, it uses LSTM networks to both predict and forecast failures and provide valuable insights to maintainers of rolling stock equipment. Our solution allows achieving over 99% of accuracy both for prediction and forecasting tasks, outperforming other ML models and techniques in the predictive maintenance literature. Further, comparing the error in the prediction and forecasting tasks, in the best case, we attained a *false alarm rate* of ∼0.4% and a mean absolute error in the order of 10^{-4}, respectively, which are very promising comparing to the existing study. The industry partner validated the proposed methodology against the real target system acknowledging the proper assumptions and approaches.

The remainder of the paper is organized as follows. Section 2 summarizes the state-of-the-art ML applications for predictive maintenance in various industry areas. Section 3 shows a theoretical background regarding the target system and the LSTM approach used for prediction. Section 4 depicts the proposed framework for dataset analysis and fault prediction. Section 5 analyzes the results obtained. Section 6 concludes the paper.

2. Related work

Emerging maintenance strategies focus on forecasting failures to plan the necessary actions with timing that maximizes the use of the systems, while not compromising functionality. Unlike the traditional experience-based diagnostic approach of maintainers and technicians, this recent paradigm is based on the ability to document and analyze knowledge and data related to anomalies and failures, minimizing human errors and standardizing methods and procedures.

The two broad categories of predictive maintenance are *knowledge-based* and *data-driven*. The former combines the knowledge of maintainers and designers with analysis techniques such as FMECA (Failure Mode, Effects, and Criticality Analysis) and RAMS (Reliability, Availability, Maintainability, and Safety). The latter is based on data processing using Artificial Intelligence and Machine Learning algorithms.

In the railway domain, tele-diagnostic systems have been introduced for some decades. They allow sensing of the operating conditions of the many on-board train subsystems and send data in real time to a control room to detect train anomalies. This laid the ground for new prognostic systems (Corfiati, Dalla Chiara, & Galfrè, 2011), able not merely to provide maintainers with information about the occurrence of anomalies, but to predict them to prevent unavailability.

For predicting failures and estimating the RUL of railway systems, statistical techniques have first been investigated. *Guclu et al.* exploited AutoRegressive-Moving-Average (ARMA) to predict the degradation of railway turnout systems (Guclu, Yilboga, Eker, Camci, & Jennions, 2010); their experiments target ten turnout systems, and the results show a Root Mean Square Error (RMSE) for the prediction equal to 0.65. Particle Filter-based techniques (Jouin, Gouriveau, Hissel, Péra, & Zerhouni, 2016) have been used in Mishra, Odelius, Thaduri, Nissen, and Rantatalo (2017) to predict the RUL of ballasted railway track systems, in a window of the failure prediction as long as 18 months with a confidence interval of 98%. While classical statistical techniques can provide good prediction ability, they do not easily identify hidden variations in the relationships between the data, which instead represent the strengths of ML techniques and the reason why they are increasingly used in predictive maintenance tasks.

More recently, ML techniques for anomaly detection, diagnosis, or prognosis of railway rolling stock subsystems have been proposed and experimented with in several studies. *Barmada et al.* proposed Support Vector Machine (SVM) for fault detection of current arcs in the pantograph system (Barmada, Raugi, Tucci, & Romano, 2013). *Fink et al.* used Deep Belief Networks (DBN) to predict operational disruptions caused by rail vehicle door systems (Fink, Zio, & Weidmann, 2015); their technique achieves an accuracy in the prognosis of 96%. Also, *Yin et al.* leveraged DBN to perform the automatic fault diagnosis of vehicle on-board equipment in high-speed trains (Yin & Zhao, 2016). Recently, *Yokouchi et al.* proposed an LSTM-based system for anomaly detection of railway vehicle air-conditioning unit (Yokouchi & Kondo, 2021), while *Hu et al.* designed an LSTM autoencoder in order to estimate RUL of turbofan engines (Hu & Dai, 2022).

Few studies in the literature target rail traction (sub)systems. *Zhang et al.* adopted Bayesian Networks (BNs) on features extracted with Restricted Boltzmann Machines (RBM) to estimate the RUL of the traction converter of China Railway high-speed trains (Zhang, Wang, Lu, & Jiang, 2019). The authors focused on the DC-link circuit in the CRH2 traction system, showing accuracy in predicting the degradation of the capacities and resistances in the circuit by about 98%. *Zhu et al.* evaluated the degradation of the CRH2 train traction system (Zhu, Zhang, Lu, & Jiang, 2019). They applied the *seq2seq* framework (Google Inc., 2020) in the encoding and decoding phases within an LSTM model; their results show accuracy in the prediction of 90%. Both these studies used datasets of train operational data generated by a hardware-in-loop system.

Table 1 summarizes the related studies.

Table 1
Failure diagnosis and prediction studies in the railway domain.

Authors	Technique	Target system	Results
Mishra et al. (2017)	Particle filter-based	Railway switches	Better performance compared to regression models with an 18-month RUL prediction for 4 railway switches
Guclu et al. (2010)	ARMA	Railway turnout	RUL turnout system with RMSE of 0.65
Barmada et al. (2013)	SVM	Pantograph	Fault detection with 90% of accuracy
Fink et al. (2015)	DBN	Rail vehicle door	Predict operational disruptions caused by rail vehicle door systems with an accuracy of 96%
Yin and Zhao (2016)	DBN	Vehicle on-board equipment	Diagnose faults of vehicle on-board equipment in high-speed trains with accuracy up to 95%
Yokouchi and Kondo (2021)	LSTM	Air-conditioning unit	Detection of anomalies up to 1.5 months before they become apparent
Hu and Dai (2022)	LSTM autoencoder	Turbofan Engine	RUL of the turbofan engine with RMSE = 0.14
Zhang et al. (2019)	RBM - BN	Traction	Better RUL matching compared to BN- and DBN-based prediction methods
Zhu et al. (2019)	LSTM seq2seq	Traction	RUL of the traction system with an error of 90%

Since the surveyed article focused on a different target system, it would be unfair to compare the existing approaches with the results obtained in our experimentation. However, we can still analyze the orders of magnitude between metrics used for evaluating the performance of ML employed in state-of-the-art studies, when available. Given that, our methodology allows achieving very high accuracy (both for prediction and forecasting tasks), over 99%, which is greater than the maximum, i.e., 96%, obtained by state-of-the-art studies. Also comparing the error in the prediction and forecasting tasks, in the best case, we attained a *false alarm rate* of ∼0.4% and a mean absolute error in the order of 10^{-4}, respectively, which are very promising comparing to the existing study. Despite this, we remark that our study is not merely about applying an ML algorithm to a specific case study. Instead, we provide a deep learning-based methodology that allows properly treating datasets from rolling stock equipment, by taking into account peculiarities like gradually changing data, the train routes, to improve accuracy in prediction and forecasting tasks. To the best of our knowledge, our study is the first attempt at leveraging LSTM machine learning models to both predict and forecast failures of a critical train traction subsystem. The proposed methodology combines LSTM models with an advanced data pre-processing algorithm, with the final aim of supporting the optimization of maintenance activities and reducing train unavailability. It has experimented on real train operational data provided by the industry partner, producing remarkable results acknowledged by the domain experts.

3. Background

In this section, we introduce the reader to the basic notions to understand the proposed methodology. Section 3.1 provides an overview of the traction converter cooling system, which is the target we focus on in the experimental section. This is fundamental to point out which are the key features to be considered when the rail vehicle traction is properly working and reveals failures. Due to the *gradually change* nature of key features we leveraged the LSTM machine learning model that easily fit such a behavior. We discuss in Section 3.2 the basics for LSTM networks, which are one of the fundamental blocks within the proposed methodology.

3.1. Traction converter cooling system

In this section, we provide a brief background on the traction converter cooling system, which is the target on which we apply the proposed methodology described in Section 4.

In the railway domain, the need for efficient cooling of electronic equipment is due to guarantee proper control of the temperature of semiconductors junction. A water and glycol mixture was selected as the medium for heat transfer due to its high thermal conductivity. The cooling system consists mainly of a liquid tank/expansion tank, a pump, a heat exchanger with the modules to be cooled, and some measuring and control equipment. In our system, the hydraulic circuit provides cooling for two main power inverter modules and one auxiliary module. The hydraulic circuit cools the Insulated Gate Bipolar Transistors (IGBT) as well as the resistors located on each of the three modules through a high thermal efficiency plate. The liquid is in turn cooled through a heat exchanger with forced ventilation air.

The cooling system is positioned on the roof of the train inside a converter box. Cooling air is taken from the head of the carriage (in the train direction) and is discharged on the sides, after passing through a sealed compartment, inside which transformers and inductors are located.

Fig. 1 shows the hydraulic circuit block diagram with its components and measuring equipment:

- The `WT` control unit tank is used both as a filling and compensation tank, as the fluid is subject to variations in volume depending on the temperatures reached. If anomalous overpressures are generated, the safety valve `VS` vents the system;
- The `TL1/TL2` switches control the cooling liquid level. In particular, `TL1` signals a pre-alarm for liquid refilling, in case of reduction in volume; `TL2` signals that the minimum admissible level has been reached, resulting in the shutdown of the system (excluding the converter);
- The `PM1 centrifugal pump` produces the liquid circulation at proper flow rate;
- The pressure point of the `PS pressure switch` is positioned on the delivery pipe, set to a minimum pressure value, in correspondence to which a malfunction of the pump or incorrect valve positioning is possible. Passing this threshold determines a shutdown of the plant;
- The `P1 pressure gauge` displays the pressure value present in the system;
- On the return line, the `flow sensor (SF)` connected to the `Flow Control Board FL1` (Coflu/4 Assembly) displays and controls the flow rate of the liquid circulating in the system. A minimum flow rate is set on the board, below which correct refrigeration is no longer guaranteed, determining a system shutdown. The typical causes that can determine the intervention of the minimum threshold of *FL1* are similar to those described for the *PS pressure switch*;
- In the operational phase, the `filter basket (F)` is used to retain the particles with dimensions greater than a specified threshold. The filter is ignored in the nominal operating position;
- Inside each module (external to the control unit) there are `PT100 thermometers`, set on a temperature limit value suited for the correct functioning of the semiconductors. They allow continuous temperature monitoring.

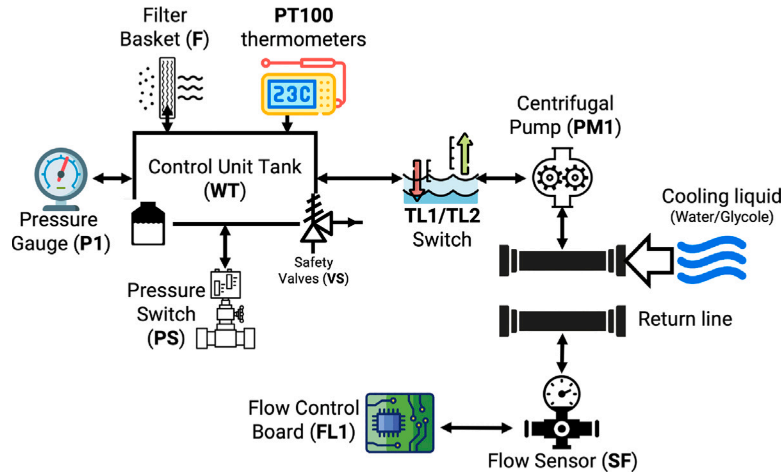

Fig. 1. Hydraulic circuit, converters unit.

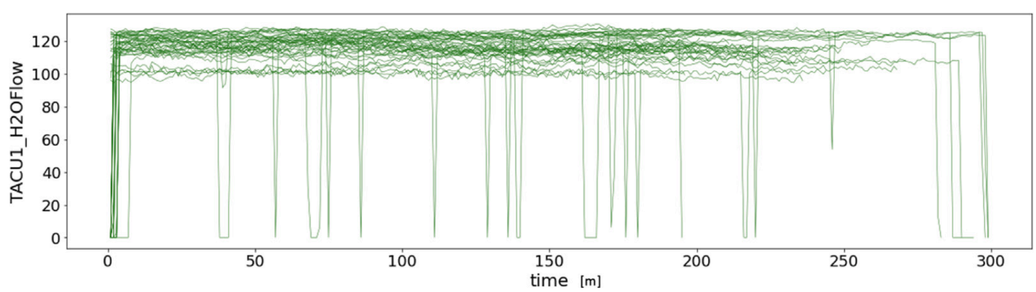

Fig. 2. An example of the `TACU H2O flow` feature signal trend over time.

For all the above components, operating conditions are monitored and data are stored in datasets, provided by the industrial partner. These can be analyzed to unveil long-term dependencies among measurements of the relevant physical dimensions within the traction system. We focus the experimentation on measurements regarding the cooling liquid level (i.e., *Status Level (TL1/TL2)*), and the flow sensor within the flow control board (i.e., *Assembly FTF (FL1)*). As suggested by the industrial experts, and more detailed in Section 4.1, these components strongly impact traction operation.

Fig. 2 shows a trend of the flow rate of the cooling liquid in a traction control unit, namely the `TACU1_H2OFlow` feature in a dataset. This trend is *gradual changing*, with some time intervals in which the signal drops to zero due to failures during operation. This behavior suggests leveraging a machine learning model, which is capable of identifying such trend characteristics. We leveraged the Long Short-Term Memory (LSTM) model to answer the research question *R1*, which is about how to properly learn long-term dependencies for gradually changing data.

3.2. Long short-term memory networks

Because of the sequential, gradually changing nature of the features in our dataset, it is important to capture in the model their dependencies over time. To this aim, we leverage a model based on Long Short-Term Memory (LSTM) networks (Gers, Schmidhuber, & Cummins, 2000; Hochreiter & Schmidhuber, 1997). These are an extension of Recurrent Neural Networks (RNNs) with an efficient gradient-based algorithm to keep the error constant, avoiding its explosion or vanishment. The main advantage concerning traditional RNNs is that LSTM networks are explicitly designed for keeping the memory of time dependencies among inputs for a long period.

Fig. 3 shows the internal structure of an LSTM cell. It consists of three gates referred to as the input gate, exit gate, and forget gate, governed by the following equations:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{6}$$

where: f_t, i_t, o_t denote, respectively, the *forget gate*, the *input gate*, and the *output gate*; W_f, W_i, W_c, and W_o denote the weight matrices related to the relative gates; U_f, U_i, U_c, and U_o denote the recurrent connections to the relative gates; σ_g, σ_c, and σ_h refer to a sigmoid function, hyperbolic tangent function, and hyperbolic tangent function, respectively; finally, $\odot$ denotes the *Hadamard product* (element-wise product).

The first step performed when data enter the LSTM cell is to select the information to store. This step is regulated by Eq. (1). The x_t data array containing the values of the features at instant t is input, along with the h_{t-1} output. The result obtained is given in input to the activation function (sigmoid σ_g) which returns a value between 0 and 1 for each element of cell state c_{t-1}, where 0 and 1 indicate if the element can be ignored or stored in time, respectively. The second step is related to the input gate and allows the LSTM to establish which

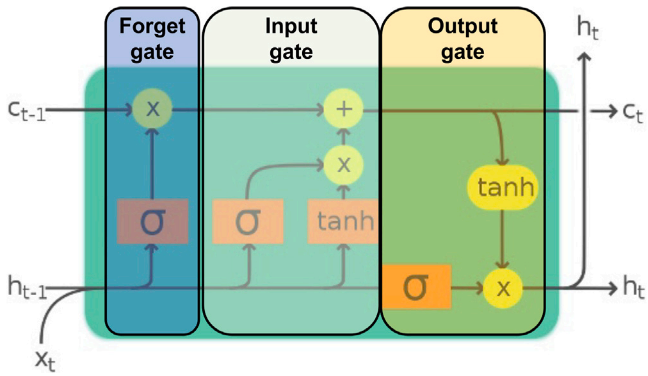

Fig. 3. Internal structure of an LSTM cell.

new information to store. This step is realized through two operations related to Eqs. (2) and (3). At the end of these operations, the state of the LSTM cell is updated through Eq. (4). Finally, the last step consists of the generation of the LSTM output through Eqs. (5) and (6) which select the part of the new state to be put in output.

In the following, we describe the methodology and show how LSTM accomplishes classification and forecasting tasks to predict traction converter cooling system failures.

4. Methodology

Because of the sequential, gradual changing nature of the features, it is important to capture dependencies among features over time. The proposed methodology, based on LSTM networks, is shown in Fig. 4. It entails the following steps:

- **Data Exploration**, data collected from various sensors in the system are analyzed to determine the relationships between the present characteristics and identify a trend over time of this latter, whose results are crucial to identify the faults within the system.
- **Data Pre-processing**: the dataset is prepared for the application of the next step of prediction. This step consists in *data cleaning*, *spike filtering*, *routes extraction*, and *sequence extraction*, described in detail in Section 4.2.
- **Fault Analysis**: failure prediction (classification) and forecasting tasks based on LSTM, for each temporal sequence after the pre-processing step.

4.1. Data exploration

The available dataset contains data from sensors and control units from the traction converter cooling system. The dataset also includes the train identifier, train location with GPS coordinates, train current speed, and the timestamp of the measurement. The dataset consists of 79,861 samples, with 131 different features, covering time series related to a fleet of 10 different trains in operation from November to December 2019. We focused on the specific period since it was indicated by our industry partner because they wanted to analyze the most problematic time interval for their train fleet. The data corresponding to the first half of November is acquired with a sampling time of 60 s, while from the second half of November until the first half of December the sampling time is about 1 s and then returns to 60 s from the second half of December. As described in the subsequent data pre-processing step (specifically, the *data cleaning* task), the processing is carried out to uniform the sampling at 60 s.

In addition to the traction converter cooling system dataset, the industry partner provided another dataset with diagnostic data about failures that occurred during train operations in the same period. This dataset contains all the diagnostic events, a brief description, the duration, and the concerned train within the fleet. The dataset is crucial to get insights into maintenance and fault predictions.

All the raw data within the dataset were collected by our industry partner by using their own network of Internet of Thing (IoT) sensors installed on-board trains, which have a direct connection to a centralized database system capable of storing a high volume of data each of them characterized by timestamp. As soon as the train completes the operation for that day, an operator triggers the function of storing recorded data in a centralized database. The database can be accessed for further elaboration. Other details on the monitoring architecture cannot be provided due to the non-disclosure agreement.

In order to explore more insights into the dataset features, this first step consists of statistical tests to identify potential trends in the dataset, as well as possible correlations between features. In the first case, the Mann–Kendall test was applied (Kendall Maurice, 1975), while in the second case, a covariance and correlation analysis was carried out (Quade, 1967).

For the case study, this first step unveiled an unknown correlation between the IGBT sensor (the `TACU1_ThIGbt2` feature) and the temperature of the cooling circuit (the `TACU3_ThH20` feature), and an interesting aspect is the detection of an anomaly when both Mann–Kendall and covariance tests failed. This result is confirmed by the comparison with the available diagnostic data. In particular, the anomalies identified are those related to the events within the *flow rate* (low or below the minimum) and *level* (low or very low) of the cooling liquid. Indeed, our industry partner acknowledged us that diagnostic events related to the flow rate (i.e., `TACU_FL1` with low severity, and `TACU_FL2` with a high severity) and the level of cooling liquid (i.e., `TACU_TL1` with low severity, and `TACU_TL2` with a high severity) are precursors of train traction failures. Given the size of the dataset, it was decided to split it into the different routes traveled by train.

4.2. Data pre-processing

In this step, a series of operations are carried out necessary to make the dataset ready for the training of the LSTM algorithm.

4.2.1. Data cleaning

When dealing with data collected from physical sensors, data cleaning becomes the first mandatory step for their analysis. In particular, to obtain a proper dataset, we need to filter out:

- *Missing or incomplete data*: delete values within the dataset that appear as blank or as special values (e.g., Not a Number, or Null);
- *Unusual values*: eliminate unexpected values compared to the normal distribution of the dataset;
- *Incorrect data formatting*: reviewing the dataset so that all data are present in the correct format which corresponds to defining decimal values with a point and not a comma;
- *Combination of redundant values*: find the data with the same information.

As defined before, the dataset had sampled data with a different sample time, so it was necessary to standardize the sampling of the data to 60 s. We use a simple algorithm that analyzed data sampled each second and averages them on a one-minute interval.

4.2.2. Routes extraction

An *ad hoc* strategy has been implemented to extract the train routes from the dataset, as basic elements for further analysis for the deep learning model. The implemented solution removes the dependency on GPS coordinates for the train position. Indeed, this feature, within the provided dataset, can be a piece of unreliable information for the actual operation of a train for inferring train routes. Therefore, we implemented an algorithm that leverages the *stop time* of a train in a station, to select the best thresholds able to recognize the beginning of a new train route. The obtained threshold is about 10 minutes, with 63 unique train routes found.

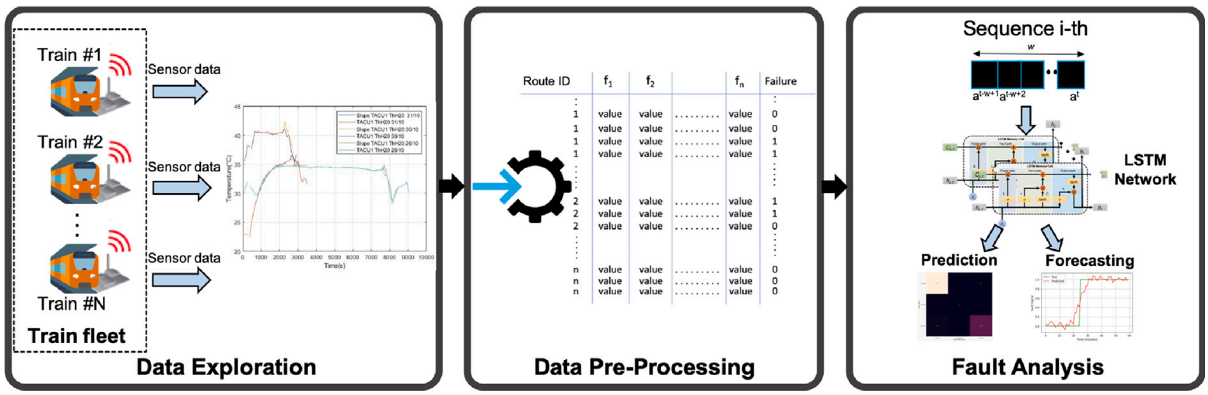

Fig. 4. The proposed failure prediction methodology.

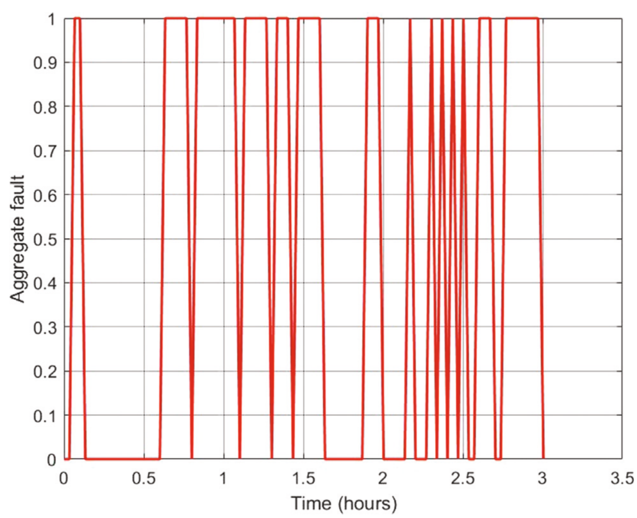

Fig. 5. Aggregate signal of diagnostic events `TACU_TL1`, `TACU_TL2`, `TACU_FL1`, `TACU_FL2` series over time for the `Milano-Bologna` train route.

4.2.3. Spike filtering

In order to implement fault prediction through LSTM networks, we need to know the error label to be assigned to each sample in the dataset. According to the provided diagnostic dataset, we add this information to the traction converter cooling system dataset. Then, a preliminary analysis of the dataset, enhanced with failures data, showed up several *spikes* that could invalidate the LSTM prediction due to the non-negligible fluctuations in the signal. For instance, Fig. 5 shows the aggregate signal about the four diagnostic events `TACU_TL1` and `TACU_TL2` events, (which denote a *low* and *very low* level of the cooling liquid), and `TACU_FL1`, `TACU_FL2` events (which denote low or below the minimum flow rate). We can notice several spikes with different duration, in which 0-signal indicates the absence of failures, while 1-signal indicates a failure of a specific duration over time according to the analyzed route.

In order to address the spike problem, we developed an algorithm (formalized in the form of pseudo-code shown in Algorithm 1) that includes, as input, the time-series of traction converter cooling system dataset enhanced with failure events, and then it evaluates, for each route, the duration of each spike to choose if that spike should be filtered out because it is most likely noise or not.

After an initial analysis of the spike distributions and their duration, a signal with a duration less than or equal to the `low_thr` (set equal to 240 s) was chosen for filtering out a spike (Algorithm 1, *line 4*); that signal will not be considered a significant diagnostic event (fault) because of its short duration. Instead, a signal that has a duration greater than or equal to `high_thr` (set equal to 800 s) is chosen as a significant fault (Algorithm 1, *line 13*).

```
Input: Traction converter cooling system dataset enhanced with
       failure events. low_thr = 240s, high_thr = 800s,
       max_spike_count = 3
Output: Traction converter cooling system dataset with spikes filtered
1  while Route is available in the route list do
2      while Diagnostic event is available in the route do
3          Compute the duration of the event;
4          if Duration ≤ low_thr then
5              Count the number of consecutive spikes (spikes' train);
6              if spikes' train size is ≤ max_spike_count then
7                  Filter the current event and the related spike's train;
8              else
9                  Merge all the spikes within the spikes' train as one
                     high signal;
10             end
11         else if Duration > low_thr and < high_thr then
12             Ignore the event;
13         else if Duration ≥ high_thr then
14             Find the first next significant failure (duration ≥
                 high_thr);
15             Count the number of spikes (signal with duration ≤
                 low_thr) between the two signals;
16             Keep high only the first spike and filter out the
                 remaining spikes in the spikes' train;
17         end
18     end
19 end
```

Algorithm 1: Algorithm pseudo-code for spike filtering.

We further refine the spike filtering technique by counting the number of consecutive spikes (*spikes' train*) between two signals (Algorithm 1, *lines 5–9*). If the number of consecutive spikes (event samples with a duration within `low_thr`), which form a *spikes' train*, is less than `max_spike_count` (set equal to 3; this choice was made based on the analysis performed previously), the algorithm filters out the diagnostic event and the related *spikes' train* (Algorithm 1, *line 7*). Otherwise, if the size of the spikes' train is greater than `max_spike_count`, the algorithm merges the diagnostic event sample within the spikes' train to obtain a single high signal (Algorithm 1, *line 9*).

In the remaining cases, according to a conservative approach, if the current diagnostic event sample has a duration greater than or equal to `high_thr`, we search for the next significant diagnostic event sample (duration ≥ 800 s in our case); then, the algorithm computes the spikes' train within these two samples and keeps high only the first spike in that train, filtering out the remaining spikes (Algorithm 1, *lines 14–18*). Finally, when the current diagnostic event sample has

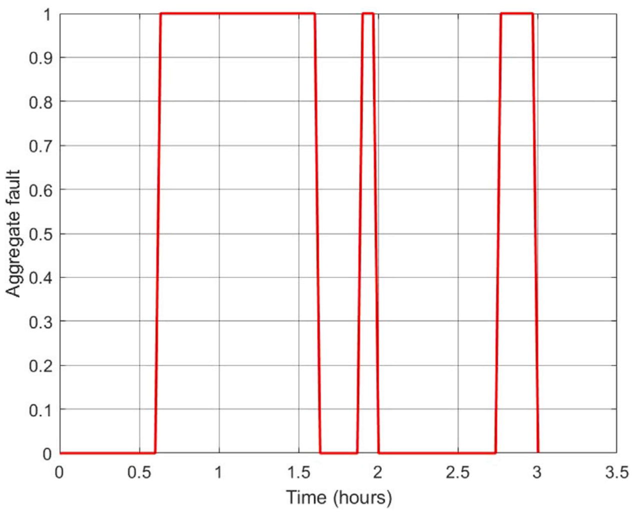

Fig. 6. Aggregate signal of diagnostic events after application of spike filtering algorithm.

a duration between `low_thr` and `high_thr`, the algorithm simply ignores (keeps high) the signal (Algorithm 1, *lines 11–12*).

Fig. 6 shows the aggregate signal of targeted diagnostic events (`TACU_TL1`, `TACU_TL2`, `TACU_FL1`, `TACU_FL2`) over time after applying the proposed algorithm.

4.2.4. Sequence extraction

In order to explore the time dependencies within the features periodically collected for each sensor, we extract the characteristic sequences on specific *time windows* (TW). Let w and a^t be the time window size and the set of features $(f_1, f_2...f_n)$ at time t, respectively. The developed model aims to predict health status at time $t+1$ $(Hs(t+1))$ considering the sequence $(a^{t-w+1}, \ldots, a^{t-1}, a^t)$. For each a^t, the health status $Hs(t)$ is defined, and the feature sequence at time t is extracted considering the $w-1$ previous samples. As a result, we obtain each sequence that consists of a bi-dimensional array of size $w \times n$, where n is the number of features considered. The result of this step is a sequence-based dataset. More specifically, the obtained dataset consists of bi-dimensional arrays, each associated with a health level representing the traction converter cooling system's health condition between two consecutive samples (i.e., a^t and a^{t+1}).

4.3. Fault analysis

This step aims at performing both a multiclass classification task and a forecasting task. More in detail, three severity levels, which have been suggested by our industry partner, have been established to categorize the components under analysis in terms of their level of severity. In the classification task, each feature sequence is assigned to one of the classes `Good`, `Minor`, `Major`. In particular, they indicate respectively: `Good`, a normal state of the component; `Minor` a warning related to the flow rate (low or below the minimum) of the cooling liquid; `Major`, a very low level of the cooling liquid.

In the forecasting task, each feature sequence extracted with a specific TW is assigned as a label that is the trend of the features in the next sequence (prediction window).

The input to each LSTM layer is a three-dimensional data structure of size $z \times w \times n$, where: z, is the total number of sequences (or the batch size at each iteration); w is the size of each sequence – i.e., the size of a time window in terms of time steps; n is the total number of features describing each time step. The network has two stacked LSTM layers with 256 units, followed by a single dense layer.

5. Experimental analysis and results

This section aims to evaluate the effectiveness of our approach applied in a real case study in the railway domain. We test the prediction performance of the proposed approach on the dataset including samples from sensors and control units from the traction converter cooling system. As described in Section 4.1, the dataset consists of 79,861 samples related to 10 different trains acquired with a time step of 60 s and 1 s. More in detail, each sample consists of 100 measures of sensors and control units from the traction converter cooling system. In particular, we analyze the performance provided by the proposed approach both for classification and forecasting tasks.

In all experiments, 10-fold cross-validation (CV) has been performed to assess generalization ability. This is of crucial importance for a trustworthy comparison of the performance of different models, avoiding the use of sequences from the same train both in the training and in the evaluation phase. For each CV repetition, 8 folds are used as the training set, 1 as the validation set, and 1 as the test set. Moreover, for each CV repetition, ADAM (Kingma & Ba, 2014) has been used as the optimizer by setting the hyper-parameters β_1, β_2, and the *learning rate*, respectively equal to 0.9, 0.999, and 10^{-4}. The loss function used to train the classification model is $categorical\_crossentropy$, for forecasting model $mean\_squared\_error$. The number of training epochs had been fixed to 150 and 100 for classification and forecasting, respectively. Firstly, we evaluate the performance of both the effectiveness and efficiency of the proposed framework in learning the model. The LSTM-based network design comprised 880,158 parameters (neurons), leading to a model storage size of approximately 9.3 MB. Furthermore, the average inference time required to generate a prediction was measured, with the results indicating an average duration of approximately 3 s. The values are acceptable and acknowledged by our industry partner.

5.1. Failure prediction

In order to answer the research question *R2*, we first report the results obtained for the prediction (classification) task, where each feature sequence is assigned to one of the classes `Good`, `Minor`, `Major`.

The performance of our approach is first evaluated in terms of *accuracy*. Since the distinction between good and failed signals is preserved in the labeling of the dataset, we can express the results in terms of accuracy on good sequences (ACC_G) and accuracy on failed sequences (ACC_F) — respectively, the fraction of sequences correctly classified as `Good`, and the fraction of sequence classified as the health levels (`Minor` or `Major`) as suggested by our industry partner. Furthermore, we evaluate the performance of our approach in terms of failure prediction ability, by assessing the *Failure Detection Rate* (FDR) and the *False Alarm Rate* (FAR). This is done by considering the class `Good` as *System good status*, and the classes `Minor` and `Major` as *System failed statuses*. Intuitively, *FDR* is the fraction of failed sequences that are correctly classified as failed, while *FAR* is the fraction of good sequences that are incorrectly classified as failed.

Tables 2 and 3 show the results of the classification task based on the LSTM technique. In particular, Table 2 reports the performance for different sizes of the time window used in the sequence extraction step (Section 4.2.4). We explored time window sizes from 5 to 60 minutes. As expected, given the ability of LSTM networks to learn long-distance dependencies, we obtain the best results with a time window equal to 60 minutes. For completeness, we detail the performance of our best model (TW size = 60 minutes) for each class in Table 3.

We compared our solution against *Decision Tree*, *Random Forest*, and *Multilayer Perceptron*, which are classical ML approaches often adopted in industrial domains (Angelopoulos et al., 2020; Gourisaria, Agrawal, Harshvardhan, Pandey, & Rautaray, 2021). The purpose of this comparison in addition allows us to analyze the effectiveness of our model in

Table 2
Performance values for the LSTM models obtained by varying TW size.

TW size [min.]	*Accuracy*	*FDR*	*FAR*	ACC_G	ACC_F	ACC_{Major}	ACC_{Minor}
5	98.67%	98.58%	1.98%	99.56%	95.19%	86.66%	95.41%
10	98.84%	98.96%	1.70%	99.80%	95.08%	92.85%	95.13%
15	98.86%	99.24%	1.23%	98.82%	99.00%	94.33%	99.12%
30	99.04%	99.32%	1.05%	99.02%	99.10%	97.77%	99.14%
60	99.45%	99.42%	0.35%	99.64%	98.76%	100.00%	98.73%

Table 3
Results of best model ($TW\,size = 60$ min) detailed by each class.

Metric	*Good*	*Major*	*Minor*
Accuracy	99.64%	100.00%	98.73%
Precision	99.80%	71.40%	99.00%
Recall	99.60%	100.00%	98.70%

learning the temporal patterns characterizing the dataset. In fact, these classical ML models are *sequence-independent* models, i.e., they do not leverage a sliding window for training but a single tuple. The results of the comparison are reported in Table 4. Specifically, results obtained show that the LSTM-based solution performs better, confirming that the temporal nature of the data provides useful information for the task under analysis.

5.2. Failure forecasting

According to the research question *R3*, we train the LSTM-based model in order to implement the forecasting task. The requirement set by domain experts is to be able to predict patterns of the fault signal within a fixed time window in the next future (*prediction TW*); this serves to evaluate whether it is possible to complete the next route before train departure after a stop, or maintenance is required. Table 5 shows the results of the forecasting task. The results are reported for sequence extraction window size varying between 30 and 120 minutes, while the prediction TW varies between 15 and 60 minutes. Model performance was evaluated in terms of accuracy and Mean Absolute Error (MAE). Regarding the classification task, the accuracy metric provides the percentage of correctly classified sequences. Since the fault signal assumes only the values 1 and 0, in the forecasting task the accuracy measures how much the discretized predicted signal overlaps with the real one, while MAE indicates how much the predicted fault signal diverges from the actual one.

Table 5 shows forecasting results, aggregating all low- and high-severity diagnostic events (in the *Low & High* column), only low-severity diagnostic events (i.e., the *Low* column), and only high-severity events (i.e., the *High* column) respectively. In all cases, the best results were obtained by analyzing temporal sequences with a duration of 120 minutes and predicting the pattern of the signal in the subsequent 60 minutes.

Fig. 7 shows an example of the prediction provided by our framework via LSTM for forecasting the trend of a fault signal (i.e., the aggregate signal of diagnostic events TACU_TL1, TACU_TL2, TACU_FL1, TACU_FL2). In particular, the example is related to the configuration with a sequence extraction window TW equal to 120 minutes with a prediction TW equal to 60 minutes.

5.3. Threats to validity

The proposed methodology is subject to few threats to validity, which commonly affect every data-driven experimentation and research. Specifically, as *internal validity*, we need to consider the possibility that the hyper-parameters used to train the LSTM model could not be optimal. However, we performed cross-validation to improve the generalization ability of the trained model, discovering the best hyper-parameters to achieve the highest accuracy for the target dataset. As

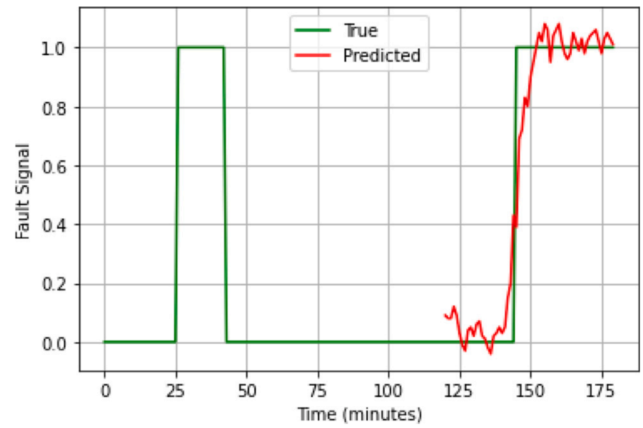

Fig. 7. Example of forecasting of a fault signal.

external validity threat, we need to consider that, by using different datasets, the provided results for prediction and forecasting tasks could be not universal. However, such a threat to validity does not invalidate the contribution since the proposed methodology can be applied seamlessly to different datasets, by a proper recalibration of chosen hyperparameters and training setup.

6. Conclusion and future work

Recent advancements in smart sensors and IT have led to continuous data collection from various on-board subsystems of railway rolling stock, enabling monitoring of mechanical, thermal, and electrical conditions, operational efficiency, and multiple other performance indicators. These new capabilities enable the planning of maintenance activities, minimizing the number and the cost of unscheduled outages. Minimizing unplanned train outages through predictive maintenance is fundamental to ensuring the reliability and stability of a transport network as a whole. In the Industry 4.0 era, via the industrial Internet of Things, the fertilization of traditional railway engineering with recent artificial intelligence techniques based on deep learning algorithms may bring important benefits.

We have proposed an LSTM-based methodology tailored for predictive maintenance in the railway domain. Specifically, the proposed framework deals with two different tasks: (i) time series classification in three severity levels as suggested by our industry partner, which can be used to assess the health status in the real-time system; (ii) time series forecasting for different time windows size to predict the system behavior in the next future, which is very useful to evaluate if it is possible to complete the next route when the train stops. We have validated the proposal against a real-world dataset from the traction converter cooling system of a train fleet maintained by the industry partner. The results show that our methodology allows achieving very high accuracy (both for prediction and forecasting tasks), over 99%, which outperforms other machine learning models proposed in the railway domain. Also comparing the error in the prediction and forecasting tasks, in the best case, we attained a *false alarm rate* of $\sim$0.4% and a mean absolute error in the order of 10^{-4}, respectively, which are very promising comparing to the existing study. These results were acknowledged by the industry partner, giving valuable insights to railway engineers in supporting predictive maintenance leading to reduced downtime of trains stopped due to breakdowns.

Future work will explore the possibility of leveraging other kinds of data-preparation techniques to improve accuracy in prediction and forecasting; further, we will exploit different, and more advanced, deep architectures for predictive maintenance and anomaly detection based on Generative Adversarial Network (Xia et al., 2022), as well as eXplainable Artificial Intelligence (XAI) techniques (Bešinović et al.,

Table 4
Results of the best model compared with classical ML models.

Methods	*Accuracy*	*FDR*	*FAR*	ACC_G	ACC_F	ACC_{Major}	ACC_{Minor}
Our solution	99.45%	99.42%	0.35%	99.64%	98.76%	100.00%	98.73%
Decision Tree	96.92%	92.16%	1.69%	98.30%	91.46%	75.80%	91.87%
Random Forest	98.18%	93.61%	0.58%	99.41%	93.28%	83.87%	93.61%
Multilayer Perceptron	93.19%	95.73%	2.05%	92.92%	94.28%	90.32%	94.38%

Table 5
Results of model for forecasting task by aggregating all the high- and low-severity diagnostic events (i.e., the *Both* column), only the low-severity diagnostic events (the *Low* column), and only the high-severity diagnostic events (*High* column).

Time Window	*Predicted TW*	Low & High		Low		High	
		Accuracy	*MAE*	*Accuracy*	*MAE*	*Accuracy*	*MAE*
30	15	99.42%	0.0197	99.92%	0.0017	99.92%	0.0014
60	30	99.38%	0.0186	99.95%	0.0013	99.95%	0.0011
120	60	99.71%	0.0184	99.97%	0.0009	99.96%	0.0008

2021). Further, we will explore the use of *contexts* (Bala & Chana, 2015) and context histories (Rosa, Barbosa, Kich, & Brito, 2015), which characterize datasets to improve predictive maintenance tasks. Indeed, contexts can be used for context prediction (da Rosa, Barbosa, & Ribeiro, 2016) and pattern similarity analysis (Dupont, Barbosa, & Alves, 2020; Filippetto, Lima, & Barbosa, 2021) in the railway domain to improve failure root cause analysis. Finally, we are planning to use the proposed framework in other kinds of subsystems within the train fleet.

CRediT authorship contribution statement

Luigi De Simone: Methodology, Software, Validation, Formal analysis, Review editing, Investigation, Data curation, Writing – original draft, Visualization. **Enzo Caputo:** Software, Validation, Data curation, Writing. **Marcello Cinque:** Methodology, Review editing, Writing. **Antonio Galli:** Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Vincenzo Moscato:** Methodology, Review editing. **Stefano Russo:** Methodology, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Guido Cesaro:** Review editing, Supervision. **Vincenzo Criscuolo:** Review editing, Supervision, Project administration, Funding acquisition. **Giuseppe Giannini:** Review editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This study was partially supported by the R&D project of the multiregional investment program "REINForce: REsearch to INspire the Future" (CDS000609) with Hitachi Rail STS S.p.A., funded by the Italian Ministry for Economic Development (MISE), and it in part was carried out within the MOST – Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023). The work by L. De Simone was supported by European Union - FSE-REACT-EU, PON Research and Innovation 2014–2020 DM1062/2021 contract number 18-I-15350-6. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., et al. (2020). Tackling Faults in the Industry 4.0 Era – A Survey of Machine-Learning Solutions and Key Aspects. *Sensors*, *20*(1), 109.
Aydin, O., & Guldamlasioglu, S. (2017). Using LSTM networks to predict engine condition on large scale data processing framework. In *2017 4th international conference on electrical and electronic engineering* (pp. 281–285). IEEE.
Bai, L., Liu, R., Wang, F., Sun, Q., & Wang, F. (2017). Estimating railway rail service life: A rail-grid-based approach. *Transportation Research Part A: Policy and Practice*, *105*, 54–65.
Bala, A., & Chana, I. (2015). Intelligent failure prediction models for scientific workflows. *Expert Systems with Applications*, *42*(3), 980–989.
Barmada, S., Raugi, M., Tucci, M., & Romano, F. (2013). Arc detection in pantograph-catenary systems by the use of support vector machines-based classification. *IET Electrical Systems in Transportation*, *4*(2), 45–52.
Basak, S., Sengupta, S., & Dubey, A. (2019). Mechanisms for Integrated Feature Normalization and Remaining Useful Life Estimation Using LSTMs Applied to Hard-Disks. In *2019 IEEE international conference on smart computing* (pp. 208–216). IEEE.
Bešinović, N., De Donato, L., Flammini, F., Goverde, R. M., Lin, Z., Liu, R., et al. (2021). Artificial intelligence in railway transport: Taxonomy, regulations, and applications. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 14011–14024.
Cachada, A., Barbosa, J., Leitño, P., Gcraldcs, C. A., Deusdado, L., Costa, J., et al. (2018). Maintenance 4.0: Intelligent and predictive maintenance system architecture. *1*, In *2018 IEEE 23rd international conference on emerging technologies and factory automation* (pp. 139–146). IEEE.
Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., & Li, X. (2020). Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Industrial Electronics (TIE)*, *68*(3), 2521–2531.
Corfiati, M., Dalla Chiara, B., & Galfrè, M. (2011). Evolution lines in the maintenance of rolling stock. *Ingegneria Ferroviaria*, *9*, 751.
De Santo, A., Galli, A., Gravina, M., Moscato, V., & Sperli, G. (2022). Deep Learning for HDD Health Assessment: An Application Based on LSTM. *IEEE Transactions on Computers (TC)*, *71*(1), 69–80.
Dupont, D., Barbosa, J. L. V., & Alves, B. M. (2020). CHSPAM: a multi-domain model for sequential pattern discovery and monitoring in contexts histories. *Pattern Analysis and Applications*, *23*, 725–734.
Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, *183*, 240–251.
Filippetto, A. S., Lima, R., & Barbosa, J. L. V. (2021). A risk prediction model for software project management based on similarity analysis of context histories. *Information and Software Technology*, *131*, Article 106497.
Fink, O., Zio, E., & Weidmann, U. (2015). Development and application of deep belief networks for predicting railway operation disruptions. *International Journal of Performabilty Engineering*, *11*(2), 121–134.
Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471.
Google Inc. (2020). Introduction to seq2seq. URL: http://google.github.io/seq2seq.
Gourisaria, M. K., Agrawal, R., Harshvardhan, G., Pandey, M., & Rautaray, S. S. (2021). Application of machine learning in industry 4.0. In *Machine learning: theoretical foundations and practical applications* (pp. 57–87). Springer.
Guclu, A., Yilboga, H., Eker, Ö. F., Camci, F., & Jennions, I. K. (2010). Prognostics with autoregressive moving average for railway turnouts. In *Annual conference of the prognostics and health management society*. PHM Society.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
Hu, L., & Dai, G. (2022). Estimate remaining useful life for predictive railways maintenance based on LSTM autoencoder. *Neural Computing and Applications*, 1–12.
Jouin, M., Gouriveau, R., Hissel, D., Péra, M.-C., & Zerhouni, N. (2016). Particle filter-based prognostics: Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, *72*, 2–31.
Kendall Maurice, G. (1975). *Rank correlation methods*. London: Charles Griffin.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Mishra, M., Odelius, J., Thaduri, A., Nissen, A., & Rantatalo, M. (2017). Particle filter-based prognostic approach for railway track geometry. *Mechanical Systems and Signal Processing*, *96*, 226–238.

Nakhaee, M. C., Hiemstra, D., Stoelinga, M., & van Noort, M. (2019). The recent applications of machine learning in rail track maintenance: A survey. In *International conference on reliability, safety, and security of railway systems* (pp. 91–105). Springer.

NASA (2020). Prognostics center of excellence HomePage. URL: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/.

Nichenametla, A. N., Nandipati, S., & Waghmare, A. L. (2017). Optimizing life cycle cost of wind turbine blades using predictive analytics in effective maintenance planning. In *2017 Annual reliability and maintainability symposium* (pp. 1–6). IEEE.

Peres, R. S., Rocha, A. D., Leitao, P., & Barata, J. (2018). IDARTS–Towards intelligent data analysis and real-time supervision for industry 4.0. *Computers in Industry*, *101*, 138–146.

Quade, D. (1967). Rank analysis of covariance. *Journal of the American Statistical Association*, *62*(320), 1187–1200.

Rahhal, J. S., & Abualnadi, D. (2020). IOT Based Predictive Maintenance Using LSTM RNN Estimator. In *2020 International conference on electrical, communication, and computer engineering* (pp. 1–5). IEEE.

Rosa, J. H., Barbosa, J. L., Kich, M., & Brito, L. (2015). A multi-temporal context-aware system for competences management. *International Journal of Artificial Intelligence in Education*, *25*, 455–492.

da Rosa, J. H., Barbosa, J. L., & Ribeiro, G. D. (2016). ORACON: An adaptive model for context prediction. *Expert Systems with Applications*, *45*, 56–70.

Saon, S., Hiyama, T., et al. (2010). Predicting remaining useful life of rotating machinery based artificial neural network. *Computers & Mathematics with Applications*, *60*(4), 1078–1087.

Sezer, E., Romero, D., Guedea, F., Macchi, M., & Emmanouilidis, C. (2018). An Industry 4.0-Enabled Low Cost Predictive Maintenance Approach for SMEs. In *2018 IEEE international conference on engineering, technology and innovation (ICE/ITMC)* (pp. 1–8). IEEE.

Williams, N., Zander, S., & Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, *36*(5), 5–16.

Xia, X., Pan, X., Li, N., He, X., Ma, L., Zhang, X., et al. (2022). GAN-based anomaly detection: a review. *Neurocomputing*.

Yin, J., & Zhao, W. (2016). Fault diagnosis network design for vehicle on-board equipments of high-speed railway: A deep learning approach. *Engineering Applications of Artificial Intelligence*, *56*, 250–259.

Yokouchi, T., & Kondo, M. (2021). LSTM-based Anomaly Detection for Railway Vehicle Air-conditioning Unit using Monitoring Data. In *IECON 2021 - 47th annual conference of the IEEE industrial electronics society* (pp. 1–6). IEEE.

Zhang, C., Wang, C., Lu, N., & Jiang, B. (2019). An RBMs-BN method to RUL prediction of traction converter of CRH2 trains. *Engineering Applications of Artificial Intelligence*, *85*, 46–56.

Zhu, K., Zhang, C., Lu, N., & Jiang, B. (2019). Data-driven RUL prediction of high-speed railway traction system based on similarity of degradation feature. In *2019 CAA symposium on fault detection, supervision and safety for technical processes* (pp. 292–297). IEEE.