A deep attention based approach for predictive maintenance applications in IoT scenarios

Roberto De Luca, Antonino Ferraro and Antonio Galli
Department of Electrical Engineering and Information Technology (DIETI), University of Naples Federico II, Naples, Italy
Mosè Gallo
Department of Chemical, Materials and Industrial Production Engineering, University of Naples Federico II, Naples, Italy, and
Vincenzo Moscato and Giancarlo Sperlì
Department of Electrical Engineering and Information Technology (DIETI), University of Naples Federico II, Naples, Italy

Received 28 February 2022
Revised 7 July 2022
5 November 2022
3 January 2023
10 January 2023
Accepted 11 January 2023

Abstract

Purpose – The recent innovations of Industry 4.0 have made it possible to easily collect data related to a production environment. In this context, information about industrial equipment – gathered by proper sensors – can be profitably used for supporting predictive maintenance (PdM) through the application of data-driven analytics based on artificial intelligence (AI) techniques. Although deep learning (DL) approaches have proven to be a quite effective solutions to the problem, one of the open research challenges remains – the design of PdM methods that are computationally efficient, and most importantly, applicable in real-world internet of things (IoT) scenarios, where they are required to be executable directly on the limited devices' hardware.

Design/methodology/approach – In this paper, the authors propose a DL approach for PdM task, which is based on a particular and very efficient architecture. The major novelty behind the proposed framework is to leverage a multi-head attention (MHA) mechanism to obtain both high results in terms of remaining useful life (RUL) estimation and low memory model storage requirements, providing the basis for a possible implementation directly on the equipment hardware.

Findings – The achieved experimental results on the NASA dataset show how the authors' approach outperforms in terms of effectiveness and efficiency the majority of the most diffused state-of-the-art techniques.

Research limitations/implications – A comparison of the spatial and temporal complexity with a typical long-short term memory (LSTM) model and the state-of-the-art approaches was also done on the NASA dataset. Despite the authors' approach achieving similar effectiveness results with respect to other approaches, it has a significantly smaller number of parameters, a smaller storage volume and lower training time.

Practical implications – The proposed approach aims to find a compromise between effectiveness and efficiency, which is crucial in the industrial domain in which it is important to maximize the link between performance attained and resources allocated. The overall accuracy performances are also on par with the finest methods described in the literature.

Originality/value – The proposed approach allows satisfying the requirements of modern embedded AI applications (reliability, low power consumption, etc.), finding a compromise between efficiency and effectiveness.

Keywords Decision support systems, Decision making, Industry 4.0, Predictive maintenance, Deep neural networks

Paper type Research paper

Journal of Manufacturing Technology Management
Vol. 34 No. 4, 2023
pp. 535-556
Emerald Publishing Limited
1741-038X
DOI 10.1108/JMTM-02-2022-0093

1. Introduction

Currently, we are living in the Industry 4.0 era that refers to the ongoing automation of traditional manufacturing and industrial practices by using modern smart technologies, such as *internet of things* (IoT) and *artificial intelligence* (AI) (Hansen and Bøgh, 2021). In this context, an increasing integration between physical and digital systems of production environments is more and more required, allowing the collection of large amounts of data that are gathered by different and distributed smart equipment and sensors.

Generally speaking, smart sensors are particular devices which generate data regarding physical parameters (e.g. temperature, humidity or vibration speed, etc.) and can provide further functionalities from self-monitoring and self-configuration to efficiently manage complex processes (Zhang *et al.*, 2019).

The analysis of such data provides at the same time useful information about the health status of the machinery and the level of production. By applying data driven analytic approaches, it is then possible to find important results for strategic decision-making, providing advantages such as: maintenance cost reduction, machine fault reduction, spare parts inventory reduction and increased production.

Most of these benefits certainly concern maintenance procedures. This aspect is particularly critical in the industrial field because it has a strong impact on the production and availability of the offered services. Nowadays, the industry is making significant investments for equipping itself with the elements necessary for applying maintenance strategies based on gathered data.

In the literature, there are two main kinds of approaches for supporting maintenance tasks, namely *model-driven* and *data-driven* methods, but also *hybrid-driven* approaches have also become popular in recent years. Specifically, model-driven techniques require expert's strong theoretical understanding to model the behavior of equipment and its detailed degradation process (Petrillo *et al.*, 2020; Liu *et al.*, 2022). From the other hand, as mentioned above, thanks to the huge amount of information that is possible to collect, data-driven techniques are emerging as the more promising ones to detect anomalies in the operation of the machinery. Regarding *hybrid-driven* solutions, they are based on the construction of multi-domain models and the development of hybrid algorithms in order to achieve model and data fusion (Luo *et al.*, 2020; Zhang *et al.*, 2022a, b).

According to Susto *et al.* (2015), approaches for maintenance management can be further grouped into three categories: *run-to-failure* (R2F), *preventive maintenance* (PvM) and *predictive maintenance* (PdM). Here, we focused on PdM, where maintenance is performed based on the health status of the specific equipment as reported by attached sensors. Through data driven analysis techniques, it is possible to know when the machinery is going to fail and then planning maintenance procedures accordingly. Basically, maintenance operations are conducted only when necessary, without waiting for machinery to report a fault. With this strategy, companies can save costs due to unnecessary maintenance, but also increase the longevity of the machine. Nevertheless, critical PdM requires several strictly requirements such as reliability, low latency, privacy and power (Mohammadi *et al.*, 2018; Sharma *et al.*, 2019).

As in many other areas concerned with huge amount of complex data to be analyzed, approaches exploiting machine learning (ML) and deep learning (DL) techniques appear to be the best among the diverse array of modern PdM techniques (Carvalho *et al.*, 2019; Ran *et al.*, 2019; Rieger *et al.*, 2019). Such approaches usually leverage historical datasets, structured as labeled time series about equipment operations, to train a variety of regression/classification models which can then be used to predict possible failures, in terms of *remaining useful life* (RUL) estimation.

Although DL approaches have proven to be a quite effective solution to the PdM problem, one of the open research challenges remains the design of PdM methods that are

computationally efficient, and most importantly, applicable in real-world IoT scenarios, where they are required to be executable directly on the limited devices' hardware.

Indeed, one of the most important issues for deploying DL architectures in real production scenario is related to the required high computational resources, not available for current equipment micro-controllers, thus favoring cloud or edge/fog computing solutions, which efficiency is often influenced by network connectivity (Teoh *et al.*, 2023).

To overcome such problems, *embedded AI* techniques are more and more diffusing to propose efficient and not computational expensive data driven analysis approach, directly executable on devices' hardware of industrial equipment (Brandalero *et al.*, 2020).

In this paper, we propose a DL approach for PdM task, which is based on a particular and very efficient architecture. The major novelty behind the proposed framework is to leverage a *multi-head attention (MHA)* mechanism to obtain both high results in terms of RUL estimation and low memory model storage requirements, providing the basis for a possible implementation directly on the equipment hardware. The attention mechanism has gained a lot of popularity in last years for its better capacities in several analytics tasks (e.g. NLP) in terms of achieved results and model complexity than recurrent models. For this reason, our basic intuition has been to rearrange such a mechanism to analyze time-series data. The achieved experimental results on the NASA dataset show how our approach outperforms in terms of effectiveness and efficiency the majority of the most diffused state of the art techniques, thus providing a suitable solution for PdM in real scenarios.

The paper is organized as in the following. Section 2 reports the related work about PdM approaches and the related challenges, while Section 3 describes the proposed methodology together with the introduced deep architecture for PdM task. Sections 4 and 5 present the experimental protocol and the achieved results, respectively. Finally, Section 6 reports a final discussion together with some conclusions and future work.

2. Related work

PdM is one of the hottest research topics of recent years and several techniques have been developed in this area. More in detail, the spread of IoT and AI technologies has led several studies to design data-driven methodologies based on ML and DL, exploiting techniques for time series analysis and mining (see Tortorella *et al.*, 2022 and Lundgren *et al.*, 2021). In addition, a recent study proposed by Sala *et al.* (2021) underlines the features of a framework for PdM capable of jointly analyzing historical and real-time data, to make a continuous improvement of its performances.

Some of the most popular proposals are summarized in Table 1, where it is easy to note as the majority of the approaches is based on *long-short term memory* (LSTM) networks and *convolutional neural networks* (CNNs). Among the plethora of DL approaches, the most interesting ones for benchmark aims are surely those leveraging the Nasa Turbofan Engine dataset. In particular, the most recent methods that have inspired our work can be classified into two broad categories: *Recurrent models* and *Hybrid models* (see Table 2). In the following, we are describing the adopted criteria for literature section, the discussed models and how our proposal overcomes the related limitations.

2.1 Selection criteria

The literature review was conducted following the guidelines introduced by Durach *et al.* (2017) and Dekkers *et al.* (2021), which are applicable in different disciplines to achieve quality systematic research.

Specifically, several steps were applied including, first, defining the research question to be met; finding samples of the relevant literature; synthesizing the literature found and

Reference	Application domain	DL approach	Output of the model	Dataset
Chen *et al.* (2019)	Automobile	AE + DNN	Estimation of TBF of an automobile	Private
Wang *et al.* (2020)	Railway Power Equipment	LSTM with residual connections	Prediction of the next failure time	Private
Kiangala and Wang (2020)	Conveyor Motor	GAF + CNN	A class corresponding to: No Fault, Minor Fault and Critical Fault	Private
Wu *et al.* (2020)	Motor Bearing	LSTM	Health Status	NASA Bearing Dataset
Zare and Ayati (2021)	Wind Turbine	Texture Signal Images + Multichannel CNN	Health Status or a specific fault	Synthetic
Li *et al.* (2019)	Rotating Machine	SAE + LSTM	/	Private
Chen *et al.* (2020)	Rolling Bearing	CNN + Bid. GRU + Attention mechanism	RUL	Private
Garcia *et al.* (2020)	Helicopter	Time-series imaging + CAE	Anomaly detection	Airbus
Martínez-Arellano *et al.* (2019)	Milling Machine Cutter	GAF + CNN	Tool wear class	Milling Machine Dataset (PHM10)
Pinedo-Sanchez *et al.* (2020)	Rolling Bearing	GS + CNN	Tool wear class	NASA Bearing Dataset
Lu *et al.* (2020)	Hard Disk	CNN-LSTM	A class indicating whether or not an HDD is going to fail	HDD Dataset
Yang *et al.* (2020)	Hard Disk	LSTM	A class indicating whether or not an HDD is going to fail	ZTE's disk dataset
De Santo *et al.* (2022)	Hard Disk	LSTM	A class representing the health state of the HDD	HDD Backblaze dataset
Purohit *et al.* (2019)	Industrial equipment	Autoencoder	Anomaly detection	MIMII Dataset
Ferraro *et al.* (2020)	Hard Disk	GAF + CNN	A class representing the health state of the HDD	HDD Backblaze dataset

Table 1.
A summary of the most recent papers regarding deep learning models for predictive maintenance

Ref	Year	Deep learning (DL) approach
Aydemir and Acar (2020)	2020	Anomaly triggered long short-term memory (LSTM)
Listou Ellefsen *et al.* (2019)	2019	Restricted Boltzmann machine (RBM) + long short-term memory (LSTM)
Ragab *et al.* (2021)	2020	Long short-term memory (LSTM) with attention
Al-Dulaimi *et al.* (2019)	2019	Long short-term memory (LSTM) + convolutional neural network (CNN)
Falcon *et al.* (2020)	2020	Multi-head attention (MHA) + LSTM + CNN + neural turing machine (NTM)
Al-Dulaimi *et al.* (2020)	2020	Noisy bidirectional long short-term memory (BLSTM) + CNN

Table 2.
A summary of the most recent papers regarding NASA turbofan engine dataset

reporting the results. Operationally, we focused on retrieving the most relevant published works in the field of PdM with DL-based approaches in different industrial contexts, with special emphasis on those that used our own dataset for experimentation (see Table 1). Works based on *recurrent* and *hybrid* network models were the considered. Eventually, the literature study led to the answer to what are the open research challenges in the PdM field.

In detail, articles on *Scopus* and *Google Scholar* in the time interval 2017–2022 were consulted and the following keywords, also combined, were used:

> *"predictive maintenance", "deep learning", "recurrent model", "hybrid", "machine learning", "attention mechanism" and "multi-head attention".*

To be specific, based on the selection criteria defined by the keywords above, 21 articles were collected, 17 of which were retrieved from *Scopus* and four from *Google Scholar*. A screening of the abstract and methodology was performed, while for only those articles that used the *Nasa Turbofan Engine Dataset*, a full-paper screening was conducted.

2.2 Recurrent models

Aydemir and Acar (2020) recently proposed a framework composed by two principal components. The first is responsible of detecting a significant deviation from the normal (healthy) condition and successively the second part (an LSTM model) is triggered for the RUL estimation. The proposed system is based on continuously checking for an anomaly and initiating continuous RUL estimation only after anomaly is detected on streaming sensor data.

Another interesting approach based on LSTM has been designed by Sohaidan *et al.* (2021) for estimating RUL, unveiling hidden patterns through the analysis of sensors sequence information. Similarly, Hesabi *et al.* (2022) relied upon a further LSTM model with data-driven approach to predict failures based on real working conditions and dynamic loading.

An *encoder-decoder* architecture based on LSTM has been then used by Ragab *et al.* (2021), also introducing an attention mechanism to deal with very long sequence. Specifically, by focusing on one aspect of the input (text, image, etc.) while paying less attention to others, attention mechanisms help direct and enhance the training process, also unveiling the relationships between input and output Niu *et al.* (2021) (see Section 3 for more details). Likewise, Li *et al.* (2022) introduced another LSTM-based attention mechanism to improve the ability of the model to analyze a sequence of signals in survival analysis. In turn, Chen *et al.* (2021) proposed a PdM system based on LSTM network adapted on FPGA, whose aim is to jointly reduce power consumption and management cost.

One of the issues related to supervised PdM applications is the lack of an adequate amount of labeled data. To challenge this problem, Listou Ellefsen *et al.* (2019) investigated the effect of unsupervised pre-training in RUL predictions utilizing a semi-supervised setup. Specifically, in the first layer a *restricted Boltzmann machine* (RBM) was utilized as an unsupervised pre-training stage in order to automatically learn abstract features from raw unlabeled input data and to initialize the weights in a region near a good starting point before supervised fine-tuning of the whole architecture was conducted. Next, LSTM is leveraged to learn long-term dependencies. Finally, a fully connected output layer is attached to perform RUL prediction.

2.3 Hybrid models

Al-Dulaimi *et al.* (2019) proposed a *hybrid deep neural network model* (HDNN), composed by two parallel paths (LSTM and CNN) followed by a fully connected layer to combine both output to predict the target RUL. This framework uses the LSTM path to extract temporal features while simultaneously the CNN is utilized to extract spatial features. An extension of this approach has been then designed by Al-Dulaimi *et al.* (2020), introducing a dual path DL architecture which is

trained with noisy input data (noisy bidirectional LSTM - NBLSTM). They highlighted that training on noisy data, in particular using a Gaussian Noise, could improve the robustness of the model leading to significantly robust and enhanced generalization behavior.

Recently, Falcon *et al.* (2020) proposed a dual-stream architecture, which consists of a MHA and a neural Turing machine module. Briefly, the time-series are first cut into shorter windows. For the labeling of these windows, piece-wise linear degradation model was used with the maximum value of the RUL fixed to 125. The obtained windows are then fed to the MHA module, which is a mean for identifying the existing relations between different sensor data in order to reveal hidden patterns among them. The output of the MHA module is then given as input to the networks in each stream. The features extracted by the LSTMs of the first stream and by the CNN of the second one are concatenated to the augmented features computed by the NTM module. At the end, two stacked feedforward networks are used to map the extracted features to a sequence of RUL values.

Hong *et al.* (2021) presented the *ConvNet* model using a CNN-LSTM network for estimating RUL of a turbofan engine, also reducing the number of parameters, while Zhang *et al.* (2022a, b) proposed a novel bi-directional gated recurrent unit with temporal self-attention mechanism (BiGRU-TSAM) to predict RUL.

Finally, Shcherbakov and Sai (2022) proposed a hybrid multi-task DL approach that integrates the advantages of CNN and LSTM networks. The former has been used as features extractor while the latter is used to capture the long-term temporary dependency features.

2.4 Research challenges in predictive maintenance

Despite numerous efforts have been made for developing approaches for PdM in recent years, a set of drawbacks can be identified.

(1) Due to the inherently sequential nature of recurrent neural networks (RNNs), this type of model precludes parallelization within training examples.

(2) When training this type of network the length of time window feeding as input could be an issue. Indeed, RNN suffers from the vanishing and explosion gradient problems. If the sequence is too long, it becomes difficult for the model to retain information about the first timesteps when processing the last ones. Although LSTMs are a type of RNN specifically designed to solve this problem, it still remains an important issue when dealing with very long-term dependencies.

Furthermore, another disadvantage of the reported frameworks concern their complexity. Even if not explicitly explained, it is clear from the models' descriptions that they have a lot of parameters, thus they require a significant storage space. Authors often do not consider this type of complexity, although it is one of the more important aspect when deploying PdM models. Indeed, the hardware usually hosting these models is resource constraint, especially in terms of memory size and power consumption.

Therefore, it is important to design fast analytics in smaller scale platforms for critical PdM applications to satisfy different requirements, as shown in Mohammadi *et al.* (2018).

(1) reliability: relying on an Internet connection may not be a viable option;

(2) low latency: these type of applications need an immediate response: transferring data to a cloud server for analysis and returning back the response is subject to latency that could cause problems;

(3) privacy: machinery-related data may be private and therefore should not be transmitted or stored in external places;

(4) power: moving data requires more energy.

Driven by the success obtained by *transformer* models in *natural language processing* (NLP) tasks (see Vaswani *et al.* (2017) for more details), several researchers decided to investigate their use in other fields. Song *et al.* (2018) developed the *SAnD* (*simpy attend and diagnose*) architecture to deal with clinical time-series data, which employs a masked, self-attention mechanism and uses positional encoding and dense interpolation strategies for incorporating temporal order. Wu *et al.* (2020) used Transformer for time-series forecasting. This is a crucial task in many scientific and engineering disciplines because it aims to predict future trends based on historical data.

The main novelty of our work lies in the introduction of an efficient MHA based deep network for PdM tasks capable of preserving good accuracy performances with reduced times and storage.

Note that, as previously reported, there are various articles in the literature of PdM making use of the attention mechanism, but always in combination with other kind of networks (LSTMs or CNNs). In this work, however, the proposed model relies only on the attention mechanism, as explained in Section 3, thus providing an efficient solution for embedded AI applications.

3. Methodology

As we have seen from previous sections, tasks related to PdM can be modeled as: (1) regression problem, in which the main purpose is the RUL estimation of a machinery; (2) a classification problem, namely the health state prediction. Using data driven techniques, maintenance procedures can be then optimally scheduled, avoiding a downtime period due to the replacement or repair of the faulty asset.

Our aim is to design an AI model capable of estimating RUL from different types of measured equipment data. The general workflow for a RUL estimation process is:

(1) Choose the best type of RUL estimation model for the data and available system knowledge.

(2) Train the estimation model using the historical data.

(3) Using test data of the same type as historical data, estimate the RUL of the test component.

In this section, we details the adopted methodology: we first provide the PdM task definition, then we present the model architecture, with a focus on the related core part, the introduced *attention* module.

3.1 Task definition

We modeled the PdM task as a particular regression problem.

Let $X = \{x_1, x_2, \ldots, x_n\}$ a set of input samples, related to the observed RUL values for our equipment in a given n-length temporal window, the PdM task consists in learning from past historical data a particular mapping function f capable of associating to the each input sequence some output samples $Y = \{y_1, y_2, \ldots, y_m\}$ in a given m-length temporal window, representing the estimated RUL values.

Formally, we have,

$$Y = f(X, \tau) + \epsilon \tag{1}$$

τ being a set of unknown parameters, which we have to determine for our model, and ϵ consists in some error terms that are not directly observed in input data.

The PdM task goal is to estimate the function $Y = f(X, \tau)$ that most closely fits the real data. Thus, we have to chosen an AI model providing a reliable mapping function f.

3.2 Model architecture

Figure 1 shows a high-level view of the proposed model architecture for the described PdM task. The figure also describes the data analysis pipeline required for the generation of estimated RUL values.

In particular, the input consists in the historical data coming from sensors giving useful information about the conditions of the monitored machinery. Such data have a temporal component, which is crucial in detecting the degradation trend.

After processing the input data, they are fed into the model able to capture temporal dependencies between features. After setting a proper time window, the data that is gathered as input to the model, is a matrix of size (T_w, N_x), where T_w is the length of the input time window and N_x is the number of considered features. The output of the model is a real number representing the RUL of the machinery.

Specifically, the main components of the proposed architecture are.

(1) the positional encoding block to take into account the relative or absolute position of the time-steps in the input sequence;

(2) the attention module, which is composed by two sub-layers (with residual connections between them):

- the MHA block;
- a fully connected network module.

In the following, we are reporting more details about each component.

3.3 Positional encoding

Since the proposed model contains no recurrences, the model does not have any sense of position and order for each timestep. Consequently, there is the need for a way to incorporate the order of the timesteps into the model. Indeed, the order is important to capture a degradation trend in analyzed data.

The proposed solution was to add a piece of information to each timestep about its position in the sequence, and this is named *positional encoding*.

The idea is to sum to each of the input timesteps a vector of length N_x that contains information about a specific position in a time window.

Let t be the desired position in an input sequence and $\overrightarrow{p_t} \in \Re^{N_x}$ its corresponding encoding, according to Vaswani *et al.* (2017), in this work we use sine and cosine functions of different frequencies,

$$\overrightarrow{p_t}^{(2i)} = \sin\left(\frac{t}{10000^{(2i/N_x)}}\right) \tag{2}$$

$$\overrightarrow{p_t}^{(2i+1)} = \cos\left(\frac{t}{10000^{((2i+1)/N_x)}}\right) \tag{3}$$

where i is the related dimension.

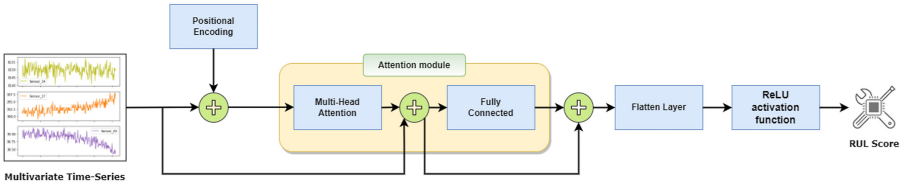


Figure 1.
Proposed AI architecture

As it can be derived from the formulas, the frequencies are decreasing along the vector dimension. Thus the wavelengths form a geometric progression from 2π to $10{,}000 \times 2\pi$. Finally, the obtained encoding for position t is summed to the t-th timestep in the input sequence.

3.4 Attention module

The proposed *attention* module is further composed by two sub-modules: the MHA sub-module and a fully connected network.

3.4.1 Multi-head attention. Attention mechanism can be described as mapping a query and a set of key-value pairs to an output. Queries, keys, values and outputs are vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is given by an arbitrary compatibility function of the query with the corresponding key. According to Vaswani *et al.* (2017), the selected compatibility function, is the *scaled dot-product attention*.

Firstly, each of the timesteps in input is linearly projected to obtain its specific query, key and value vectors of dimension d_k. Next, given a query, the dot product of the query with all keys is computed. Then, these products are divided by $\sqrt{d_k}$. Finally, the softmax function is applied to obtain the weights on the values. These weights can be seen as scores, thus they represent the importance of the values (each corresponding to one multiplied key) with respect to the value corresponding to the query. Intuitively, a subset of more important times receives high weights, while useless ones receive lower weights. At this point, weighted values are summed up.

The explained calculation is valid only when there is a single query. In practice, as we have seen, there are a number of queries equal to the number of timesteps in the selected time window, i.e. T_w. Therefore, in order to speed-up the computation, the scaled dot-product attention is computed on a set of queries of queries simultaneously, packed together into a matrix Q. If we do the same with the keys and values, the output can be expressed as,

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

MHA mechanism (Figure 2) simply repeats the above computation a number of times equal to the chosen number of heads, h. More precisely, instead of calculating a single attention function with one set of queries, keys and values, this mechanism first creates h different sets of queries, keys and values and for each of them performs the attention function in parallel.

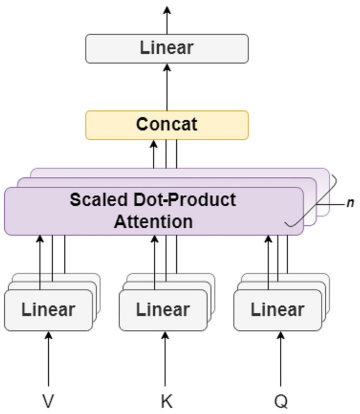


Figure 2.
Multi-head attention

The outputs of each head are concatenated and the final result is linearly projected in order to obtain the matrix of shape (T_w, N_x).

Mathematically, MHA is defined as,

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \tag{5}$$

where

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{6}$$

At a higher level of abstraction, the MHA sub-module computes a new representation of the input time window. In this representation, each timestep is enriched by the knowledge of the timesteps that precede or follow it in the sequence.

After the attention module, there is a feed forward network. This network is applied to each position separately and identically. Specifically, it consists of two linear transformations with a ReLU activation function.

Mathematically:

$$FFN(x) = \max(0, W_1x + b_1)W_2 + b_2 \tag{7}$$

As regards to the input and output dimensions, they are equal to N_x. The inner-layer has a variable dimension, given by $N_x \times FFN\_FACTOR$.

Finally, after stacking a variable number of attention modules, there are a flatten layer and a final layer containing only one neuron with a ReLU activation function.

4. Experimental evaluation

As already described, the major novelty of this work is to introduce an attention-based deep architecture for PdM. In particular, we designed it for applications requiring predictive models to be stored in memory constraint device. In order to show its validity with respect to common recurrent deep models, a comparison has been made with the most diffused architecture in terms of model's storage size and accuracy performance according to specific metrics. Specifically, we chose an LSTM network with two layers, each containing 128 units and a final Dense layer with a ReLU as activation function to perform the regression task.

Furthermore, the comparison has been realized by using Turbofan engine degradation simulation dataset [1] provided by NASA Ames Prognostics Data Repository. It is a well-known benchmark used in prognostic and health management (PHM) field. It is generated by C-MAPSS tool that simulates various degradation scenarios of the fleet of engines of the same type. It contains four sub-datasets (called FD001, …, FD004) with different operating conditions and fault modes. Each sub-dataset includes training dataset and testing dataset.

The training dataset is composed of run-to-failure sequential data collected from 21 sensors (see Saxena *et al.* (2008) for detailed information about the sensors). The engine operates normally at the beginning with certain degrees of initial wear. The sensors record the data of the engine until the fault develops to a system failure.

More in details, each row of the dataset has 26 fields.

(1) Engine ID,

(2) Cycle index,

(3) Three fields representing the operating condition of the engine,

(4) 21 sensor readings.

Test set data is different: the engine starts in an unknown deteriorated state and the readings terminates at some point prior to system failure. Therefore, the aim is to predict the RUL of each engine. For evaluation purposes, the true RUL values for the test trajectories are provided.

In the test dataset, the sensory data of the system prior to the system failure are recorded. The task is to estimate the RUL of the engine in the testing dataset. Therefore, in the testing dataset, the actual RUL of each data sample is provided to check the result of the proposed method.

4.1 Hyperparameters

In order to provide more details about the model's hyperparameters, here is reported a summary.

(1) *NUM_ENC*: the number of stacked attention modules.

(2) *NUM_HEADS*: the number of attention heads in the MHA mechanism.

(3) *KEY_DIM*: the dimension of the query and key vectors. Although it is not mandatory, in this work the value vector has this dimension.

(4) *FFN_FACTOR*: this regulates the number of neurons in the first layer of the fully connected sublayer in each of the attention modules.

However, there are other hyperparameters such as learning rate, batch size and number of epochs which are not dependent on the proposed model but affect the training stage of the model. The proposed model has been implemented in Tensorflow 2.0 using the corresponding Keras layers.

Below are summarized the steps followed in the analysis and then give more details about them.

(1) Feature selection and normalization;

(2) RUL target function definition;

(3) Time-windows creation.

4.2 Feature selection and normalization

We perform some feature selection activities, such as the elimination of constant columns. Specifically, deleted columns are: $sensor_1$, $sensor_5$, $sensor_{16}$, $sensor_{18}$ and $sensor_{19}$. Moreover, we eliminate the columns related to the operational condition, since we know it is the same across the engines.

Looking at Figure 3, we can also observe $sensor_6$ and $sensor_{10}$ are not very useful in detecting a degradation trend. For this reason, we prefer dropping these features. Obviously, this is true also for the other engines.

Finally, the ID of the engines and the cycle numbers are not used when training the model, but they are important when constructing the time windows.

After applying this features selection, we consider 14 columns.

Due to the different ranges for collected sensor measurements, a normalization step is also required to uniform the values and to provide unbiased involvement from the readings of each sensor. In particular, for each sensor we perform a min-max scaling in the range [0, 1].

4.3 RUL target function definition

Data downloaded from the NASA UCR repository cannot be used directly to train a model through supervised learning techniques because it does not contain the ground-truth. In order to solve this issue, various approaches have been proposed in the literature.

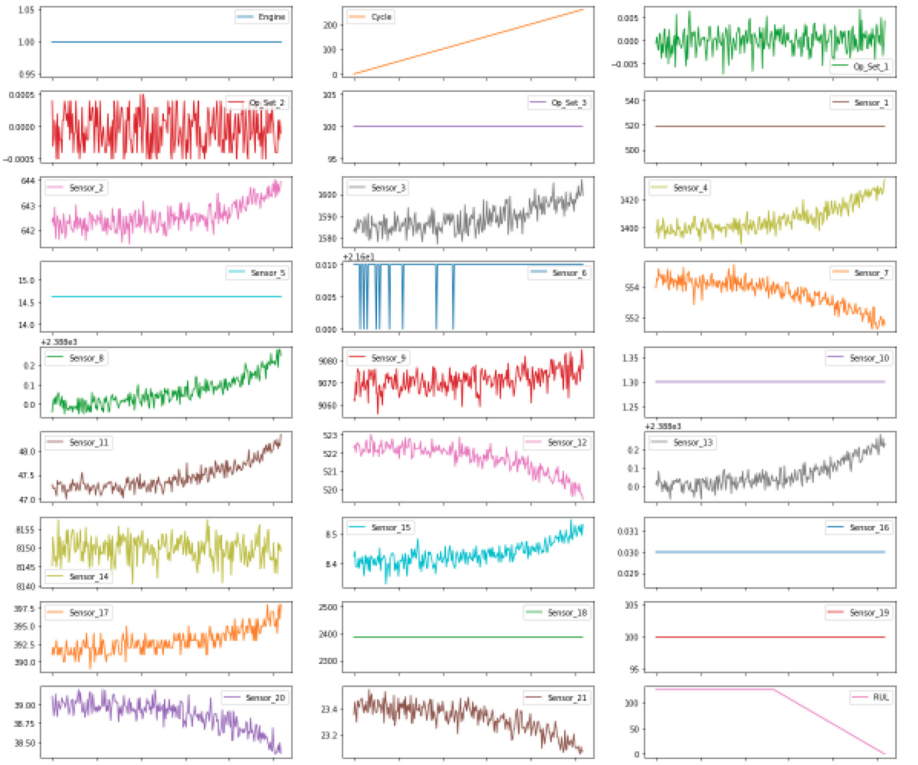


Figure 3.
Features plot of the first engine

We used the so called piece-wise linear degradation model. The main idea behind this strategy is quite simple. Because the engine failure occurred gradually, it is not appropriate to utilize the real RUL when adding the RUL label. In general, the strategy used is to establish the degradation threshold and ignore the period before the engine degrades. When the operating time reaches the degradation threshold, the engine's remaining useable life reduces monotonically. To address this issue and map this process, a piece-wise linear deterioration model was developed on the basis of Ramasso (2014) approach.

According to most of the related works, we set the clip value equal to 125.

Figure 4 shows the difference between the two main used RUL target functions.

Let us formalize this idea. Let n_i be the number of cycles for the i-th engine. Let x_i be the actual cycle of the engine. If $n_i - x_i > 125$ then the RUL value for this cycle is fixed to 125. Otherwise, it is equal to $n_i - x_i$.

4.4 Time-windows creation

To conclude the data preparation phase, since the proposed model takes a sequence of timesteps as input, there is the need to create such time windows.

We follow the sliding window method, as depicted in Figure 5. Given the window size, W, the total number of cycles, T, and the stride of the window, s, it is possible to construct $T - W - s$ time windows for each engine. For training purposes, we associate the time window the RUL value of the last timestep (cycle) it contains. In this thesis work, various time window size are used: 10, 20, 30 and the stride s is fixed to 1.

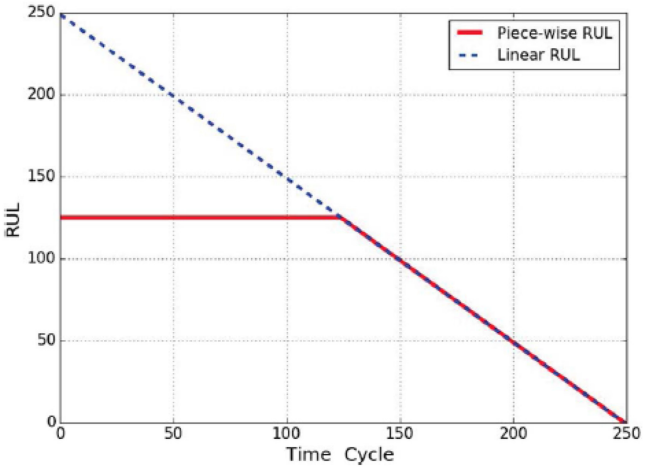


Figure 4.
Linear and Piece-wise linear degradation model

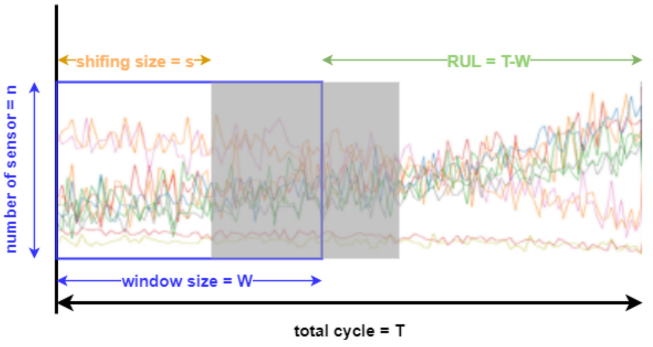


Figure 5.
Time windows creation

4.5 Performance metrics

In this section the evaluation metrics used are described. For Turbofan engine degradation dataset, we consider two objective metrics to test the performance of the model: the scoring function, and the root mean square error (RMSE), since it is a regression problem. Defining RUL'_i and RUL_i respectively the estimated and the actual RUL of the ith test engine (N in total), the RMSE can be expressed as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(RUL'_i - RUL_i\right)} \tag{8}$$

The scoring function was initially proposed in Saxena *et al.* (2008), but now is widely used in PHM applications. Defining $h_i = RUL'_i - RUL_i$, this function can be expressed by:

$$S = \sum_{i=1}^{N} s_i \tag{9}$$

where

$$s_i = \begin{cases} e^{\frac{-h_i}{13}} - 1 & if \quad h_i < 0 \\ e^{\frac{h_i}{10}} - 1 & if \quad h_i \geq 0 \end{cases} \tag{10}$$

Ideally, this function should be as lower as possible.

In Figure 6 there is a plot representing the scoring function and the RMSE. As we can see, RMSE does not make difference between an early prediction (when the estimated RUL is lower than the real) and a late one. However, in PHM it is crucial to have useful RUL predictions, i.e. predictions that makes possible to repair a machine before its failure. This characteristic is evident in the plot: as the predicted RUL is greater than the real one, the scoring function increases exponentially; if the predicted RUL is lower than the real, clearly it is an error, but not as bad as the previous case. Therefore, the scoring function increases with a lower rate.

5. Results

In this section the achieved results are reported and discussed. In particular, Table 3 shows the performance metrics of the proposed model varying the time window length. To obtain a more robust estimation, all the tests were repeated 10 times and means and standard deviations were taken. Similarly, Table 4 reports the performance metrics of the LSTM network varying the time window length.

More in details, the hyperparameters chosen for the proposed model are: *NUM_ENC* 1, *NUM_HEADS* 8, *KEY_DIM* 28 and *FFN_FACTOR* 517. Regarding the LSTM model we used two layer each with 128 units. Adam optimizer has been used to train both models, the maximum number of epochs was set to 300, batch size to 128 and learning rate to $10e-3$.

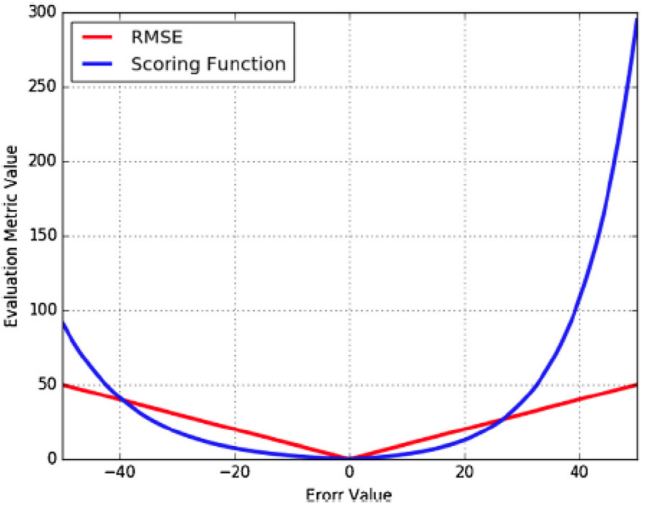


Figure 6.
Scoring function

Table 3.
Performance metrics of the proposed model varying the time window length

TW [cycles]	RMSE	Score
10	18.92 ± 0.26	1,290 ± 42
20	14.40 ± 0.21	391 ± 17
30	13.50 ± 0. 30	279 ± 23

The first consideration we can derive from looking at the results is that,as we expected, both the models benefits from the time window length increase.

Indeed, performance metrics reach their best score with a time window length fixed at 30 cycles. This behavior is quite reasonable because a larger number of samples can help in extracting a degradation pattern in the engine.

As it can be noticed by analyzing the reported results, there is no statistical difference between the two models in the score function when the time window length is equal to 20 (391 and 375) and 30 cycles (279 and 262). This difference becomes statistically significant in favor of the proposed model when considering a time window length equal to 10.

Furthermore, reported results have to be compared also looking at the model complexity. Table 5 reports the number of parameters of the models. As it can be observed, the proposed model can achieve comparable performance respect to LSTM with about 86% less parameters.

This last aspect has a strong impact in two aspect: model storage size and training time. Table 6 shows that the proposed model only requires 141 KB of memory compared to 2.5 MB required by LSTM network resulting 94.36% most efficient in terms of model storage size. Thus, it is possible to store it even if in a constrained hardware, that is usually employed in critical PdM applications.

To understand how the importance of limited memory occupancy impacts industrial contexts, we mention some published work about this topic. Concari and Bettini (2020) defined a PdM strategy on embedded plant and machinery systems, being that embedded has limited computing resources, a trade-off had to be considered between memory occupancy, running speed and accuracy. Instead, Resende *et al.* (2021) presented *TIP4.0*, a modular framework for PdM in IoT, where one of the main goals has been to offer a system that can run on hardware with limited computational power and memory.

In Table 7 there is also reported the training time of the models varying the time window length. As shown, the proposed model is 20% faster to train than the LSTM one, although performance differs slightly (about 53[*s*] on average for all window length tested case). Even if this is a limited improvement, in some real environments where the data to be processed are characterized by high dimensionality, even a small improvement in training time can make a difference, also in being able to ensure the scalability of the system (Gigoni *et al.*, 2019) with respect to dataset size.

Table 4. Performance metrics of the LSTM network varying the time window length

TW [cycles]	RMSE	Score
10	19.73 ± 0.46	1,521 ± 44
20	14.76 ± 0.28	375 ± 37
30	13.11 ± 0. 36	262 ± 20

Table 5. Number of parameters comparison

TW [cycles]	Proposed approach	Standard LSTM
10	28,233	204,929
20	28,373	204,929
30	28,513	204,929

Table 6. Models' storage size

Proposed approach	Standard LSTM
141 KB	2.5 MB

Table 8 shows a comparison with the state of the art approaches on this benchmark. Although it is not the best one, our attention-based approach is still comparable with them, especially considering the scoring function. The best results are highlighted in italic.

In Figures 7 and 8 there are reported the prediction errors respectively of the best runs of the proposed and the LSTM models.

More in details, the horizontal axis shows the test engine's ID in decreasing order with regard to their actual RUL. For example, in the test dataset, the engine with ID equal to 82 has the lowest RUL, and so on. The vertical axis shows $RULi' - RULi$, which is identified with "diff" in the plots 7 and 8, indicating the projected RUL for the i-th engine as $RULi'$.

The error is almost always less than 10 when the actual RUL has low values. However, when the actual RUL is higher, the prediction error increases in both circumstances. One probable reason for this behavior is because when the actual RUL is low, the degradation process has already begun in that engine, and hence this pattern is recognized by the models. However, when the actual RUL is very high, the engine is presumed to be in good condition, and the models cannot detect the distinction between an engine with a true RUL of 80 and one with an actual RUL of 105.

In summary, a number of useful considerations can be deduced from the analysis of all the achieved results:

(1) Our approach obtains very good accuracy performance compared with the best deep approaches in the literature, thus providing an *effective* solution for RUL estimation;

(2) Our architecture requires reduced training time and limited storage requirements, resulting in an *efficient* solution that is easily implemented on equipment hardware;

(3) Our model, despite being validated in a limited scenario, shows promising results. This suggests to us that a release in a real IoT scenario could enable the fulfillment of reliability, low latency, privacy and low power requirements (Mohammadi *et al.* (2018).)

6. Discussion and conclusions

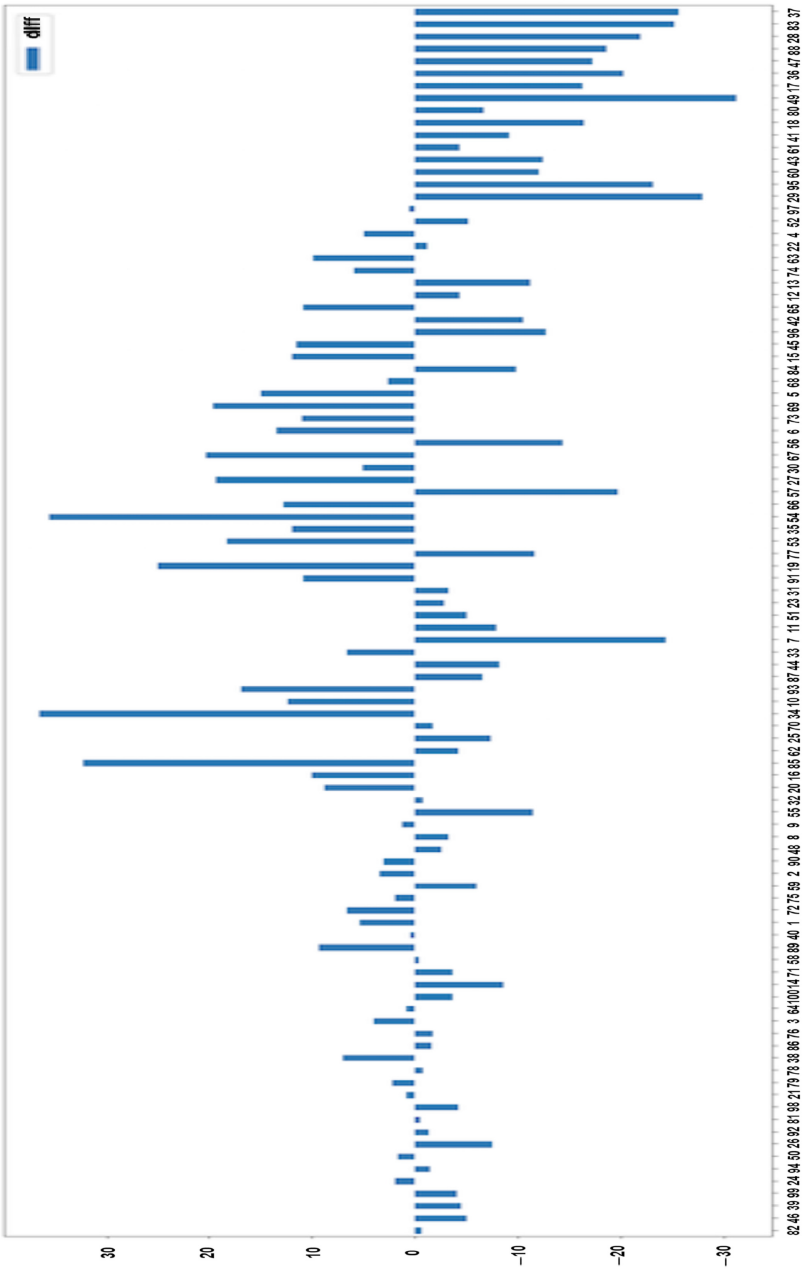
In the Industry 4.0 era, PdM plays an important role and has important managerial and practical implications, because it provides the possibility to reduce maintenance costs,

Table 7.
Comparison of training times by varying the time window (TW) between proposed model and LSTM network

TW [cycles]	Proposed approach	Standard LSTM
10	217.68 ± 4.04 [s]	272.54 ± 3.81 [s]
20	235.08 ± 1.69 [s]	290.21 ± 2.83 [s]
30	273.34 ± 2.99 [s]	323.67 ± 16.50 [s]

Table 8.
Comparison with state-of-the-art approaches

Authors	DL approach	RMSE	Score
Al-Dulaimi et al. *(2020)*	*Noisy BLSTM + CNN*	11.36 ± 0.09	226 ± 3
Ragab *et al.* (2021)	LSTM with attention	11.44	263
Listou Ellefsen *et al.* (2019)	RBM + LSTM	12.10	251
Al-Dulaimi *et al.* (2019)	LSTM + CNN	12.22 ± 0.04	288 ± 4
He (2019)	CNN + LSTM	12.46	535
Standard LSTM	LSTM	13.11 ± 0.36	262 ± 20
Proposed approach	*Attention-based*	13.50 ± 0.30	279 ± 23
Aydemir and Acar (2020)	Anomaly triggered LSTM	17.63	424
Falcon *et al.* (2020)	MHA + LSTM + CNN + NTM	/	275



Figuer 7.
Best proposed model's prediction errors

avoid failures and optimize the use of the industrial equipment. Data-driven approaches to PdM have been studied in last years, especially ones adopting ML algorithms. Thanks to these techniques, it is now possible estimate the RUL of the machinery just analyzing related past operational data.

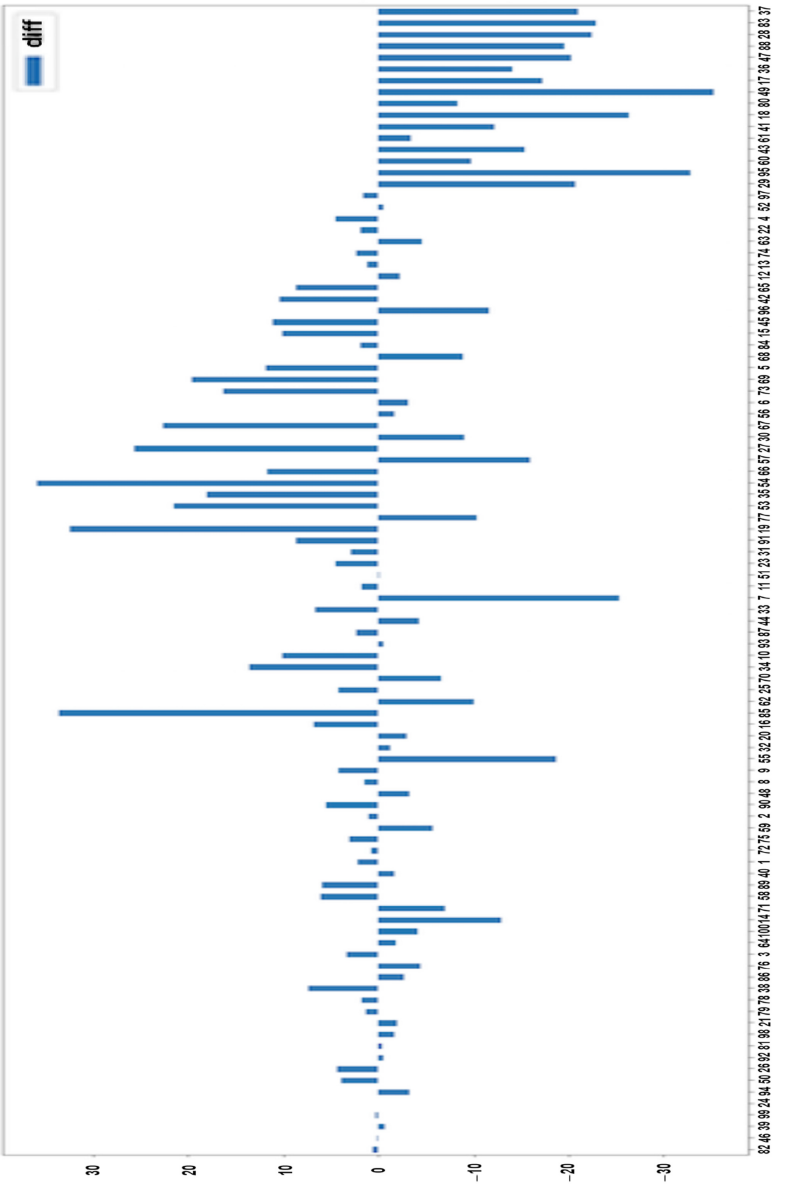


Figure 8.
Best LSTM model's prediction errors

One of most recent trend is using DL models for PdM purposes because of their state-of-the-art results in fields such as computer vision and NLP are easily extensible for PdM task. The first attempts have shown promising results.

However, proposed models often are too complex and not well suited for critical applications. Indeed, in those cases, trained models are hosted in resource constrained hardware, especially in terms of memory size and power consumption. While in general it could be possible to process

data in a cloud environment, this is not desirable for such applications requiring high reliability, low latency, data privacy and low energy consumption.

In this work, a light attention-based model has been proposed to deal with the exposed problem. The attention mechanism has gained a lot of popularity in last years for its better capacities in NLP tasks in terms of achieved results and model complexity than recurrent models. For this reason, it has been rearranged in a novel manner in this work to analyze time-series data.

In order to validate the proposal, the well-known Turbofan engine degradation dataset provided by NASA has been used. In addition to making a comparison with the best latest methods on this dataset in the literature, a spatial and temporal complexity comparison with a standard LSTM model was also made. The results show there is no significant difference in terms of RMSE and a PHM scoring function with such a recurrent model. However, the proposed model has far fewer parameters, its storage size is much lower than the LSTM's one and it is also faster in the training stage. A trade-off between efficiency and effectiveness was thus achieved, which is of paramount importance in industrial contexts where the relationship between performance obtained and resources allocated is to be optimized (Chen *et al.*, 2021; Markiewicz *et al.*, 2019). In addition, the overall accuracy performances are comparable with the best techniques of the literature.

Summarizing, the achieved experimental results showed how the proposed approach enables it to meet the requirements of modern embedded AI applications, with obvious benefits for smart manufacturing systems where the requirements of reliability, low latency, privacy and low power are imperative and important implications from a management point of view in order to optimize the operation of a production line.

Possible future works could concern the further investigation of the attention mechanism applied to different PdM applications.

Moreover, it would be interesting to inspect what actually the model learns (i.e. gives more attention) in order to provide a sort of explanations using *explainable artificial intelligence* (XAI) tools.

Note

1. https://data.nasa.gov/Aerospace/Turbofan-engine-degradation-simulation-data-set/vrks-gjie

References

Al-Dulaimi, A., Zabihi, S., Asif, A. and Mohammadi, A. (2019), "A multimodal and hybrid deep neural network model for remaining useful life estimation", *Computers in Industry*, Vol. 108, pp. 186-196.

Al-Dulaimi, A., Zabihi, S., Asif, A. and Mohammed, A. (2020), "Nblstm: noisy and hybrid convolutional neural network and blstm-based deep architecture for remaining useful life estimation", *Journal of Computing and Information Science in Engineering*, Vol. 20 No. 2, 021012.

Aydemir, G. and Acar, B. (2020), "Anomaly monitoring improves remaining useful life estimation of industrial machinery", *Journal of Manufacturing Systems*, Vol. 56, pp. 463-469.

Brandalero, M., Ali, M., Le Jeune, L., Hernandez, H.G.M., Veleski, M., da Silva, B., Lemeire, J., Van Beeck, K., Touhafi, A., Goedemé, T., Mentens, N., Göhringer, D. and Hübner, M. (2020), "Aitia: embedded ai techniques for embedded industrial applications", *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, IEEE, pp. 1-7.

Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., Francisco, R.da P., Basto, J.P. and Alcalá, S.G.S. (2019), "A systematic literature review of machine learning methods applied to predictive maintenance", *Computers andi Industrial Engineering*, Vol. 137, 106024.

Chen, C., Liu, Y., Sun, X., Cairano-Gilfedder, C.D. and Titmus, S. (2019), "Automobile maintenance prediction using deep learning with gis data. Procedia CIRP", *52nd CIRP Conference on Manufacturing Systems (CMS)*, Ljubljana, Slovenia, June 12-14, 2019, Vol. 81, pp. 447-452.

Chen, Y., Peng, G., Zhu, Z. and Li, S. (2020), "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction", *Applied Soft Computing*, Vol. 86, 105919.

Chen, J., Hong, S., He, W., Moon, J. and Jun, S.-W. (2021), "Eciton: very low-power lstm neural network accelerator for predictive maintenance at the edge", *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*, IEEE, pp. 1-8.

Concari, C. and Bettini, G. (2020), "Embedded implementation of rainflow-counting for on-line predictive maintenance", in *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, IEEE, pp. 981-988.

De Santo, A., Galli, A., Gravina, M., Moscato, V. and Sperlì, G. (2022), "Deep learning for hdd health assessment: an application based on lstm", *IEEE Transactions on Computers*, Vol. 71 No. 1, pp. 69-80.

Dekkers, R., Carey, L. and Langhorne, P. (2021), *Making Literature Reviews Work: A Multidisciplinary Guide to Systematic Approaches*, Springer.

Durach, C.F., Kembro, J. and Wieland, A. (2017), "A new paradigm for systematic literature reviews in supply chain management", *Journal of Supply Chain Management*, Vol. 53 No. 4, pp. 67-85.

Falcon, A., D'Agostino, G., Serra, G., Brajnik, G., Tasso, C. and Kessler, F.B. (2020), "A dual-stream architecture based on neural turing machine and attention for the remaining useful life estimation problem", *PHM Society European Conference*, Vol. 5 No. 1, p. 10, ISO 690.

Ferraro, A., Galli, A., Moscato, V. and Sperlí, G. (2020), "A novel approach for predictive maintenance combining gaf encoding strategies and deep networks", *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*, Nadi, Fiji, pp. 127-132, doi: 10.1109/DependSys51298.2020.00027.

Garcia, G.R., Michau, G., Ducoffe, M., Gupta, J.S. and Fink, O. (2020), "Time series to images: monitoring the condition of industrial assets with deep learning image processing algorithms", *arXiv Preprint*, arXiv:2005.07031.

Gigoni, L., Betti, A., Tucci, M. and Crisostomi, E. (2019), "A scalable predictive maintenance model for detecting wind turbine component failures based on scada data", in *2019 IEEE Power andi Energy Society General Meeting (PESGM)*, pp. 1-5.

Hansen, E.B. and Bøgh, S. (2021), "Artificial intelligence and internet of things in small and medium-sized enterprises: a survey", *Journal of Manufacturing Systems*, Vol. 58, pp. 362-372.

He, J.L.X.L.D. (2019), "A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction", *IEEE Access*, Vol. 7, pp. 75464-75475.

Hesabi, H., Nourelfath, M. and Hajji, A. (2022), "A deep learning predictive model for selective maintenance optimization", *Reliability Engineering andi System Safety*, Vol. 219, 108191.

Hong, C.W., Ko, M.-S. and Hur, K. (2021), "Convnet-based remaining useful life prognosis of a turbofan engine", *2021 IEEE 4th International Conference on Knowledge Innovation and Invention (ICKII)*, IEEE, pp. 190-193.

Kiangala, K.S. and Wang, Z. (2020), "An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment", *IEEE Access*, Vol. 8, pp. 121033-121049.

Li, Z., Li, J., Wang, Y. and Wang, K. (2019), "A deep learning approach for anomaly detection based on sae and lstm in mechanical equipment", *The International Journal of Advanced Manufacturing Technology*, Vol. 103 No. 1, pp. 499-510.

Li, X., Krivtsov, V. and Arora, K. (2022), "Attention-based deep survival model for time series data", *Reliability Engineering andi System Safety*, Vol. 217, 108033.

Listou Ellefsen, A., Bjørlykhaug, E., Æsøy, V., Ushakov, S. and Zhang, H. (2019), "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture", *Reliability Engineering andi System Safety*, Vol. 183, pp. 240-251.

Liu, P., Wang, L., Ranjan, R., He, G. and Zhao, L. (2022), "A survey on active deep learning: from model driven to data driven", *ACM Computing Surveys (CSUR)*, Vol. 54 No. 10s, 221, doi: 10.1145/3510414.

Lu, S., Luo, B., Patel, T., Yao, Y., Tiwari, D. and Shi, W. (2020), "Making disk failure predictions smarter!", *FAST*, pp. 151-167, ISO 690.

Lundgren, C., Bokrantz, J. and Skoogh, A. (2021), "A strategy development process for smart maintenance implementation", *Journal of Manufacturing Technology Management*, Vol. 32 No. 9, pp. 142-166.

Luo, W., Hu, T., Ye, Y., Zhang, C. and Wei, Y. (2020), "A hybrid predictive maintenance approach for cnc machine tool driven by digital twin", *Robotics and Computer-Integrated Manufacturing*, Vol. 65, 101974.

Markiewicz, M., Wielgosz, M., Bocheński, M., Tabaczyński, W., Konieczny, T. and Kowalczyk, L. (2019), "Predictive maintenance of induction motors using ultra-low power wireless sensors and compressed recurrent neural networks", *IEEE Access*, Vol. 7, pp. 178891-178902.

Martínez-Arellano, G., Terrazas, G. and Ratchev, S. (2019), "Tool wear classification using time series imaging and deep learning", *The International Journal of Advanced Manufacturing Technology*, Vol. 104 No. 9, pp. 3647-3662.

Mohammadi, M., Al-Fuqaha, A., Sorour, S. and Guizani, M. (2018), "Deep learning for iot big data and streaming analytics: a survey", *IEEE Communications Surveys Tutorials*, Vol. 20 No. 4, pp. 2923-2960.

Niu, Z., Zhong, G. and Yu, H. (2021), "A review on the attention mechanism of deep learning", *Neurocomputing*, Vol. 452, pp. 48-62.

Petrillo, A., Picariello, A., Santini, S., Scarciello, B. and Sperlí, G. (2020), "Model-based vehicular prognostics framework using big data architecture", *Computers in Industry*, Vol. 115, 103177.

Pinedo-Sanchez, L.A., Mercado-Ravell, D.A. and Carballo-Monsivais, C.A. (2020), "Vibration analysis in bearings for failure prevention using cnn", *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 42 No. 12, pp. 1-17.

Purohit, H., Tanabe, R., Ichige, K., Endo, T., Nikaido, Y., Suefusa, K. and Kawaguchi, Y. (2019), "Mimii dataset: sound dataset for malfunctioning industrial machine investigation and inspection", *arXiv Preprint*, arXiv:1909.09347.

Ragab, M., Chen, Z., Wu, M., Kwoh, C.-K., Yan, R. and Li, X. (2021), "Attention-based sequence to sequence model for machine remaining useful life prediction", *Neurocomputing*, Vol. 466, pp. 58-68.

Ramasso, E. (2014), "Investigating computational geometry for failure prognostics in presence of imprecise health indicator: results and comparisons on c-mapss datasets", *PHM Society European Conference*, Vol. 2.

Ran, Y., Zhou, X., Lin, P., Wen, Y. and Deng, R. (2019), "A survey of predictive maintenance: systems, purposes and approaches", *arXiv Preprint*, arXiv:1912.07383.

Resende, C., Folgado, D., Oliveira, J., Franco, B., Moreira, W., Oliveira-Jr, A., Cavaleiro, A. and Carvalho, R. (2021), "Tip4. 0: industrial internet of things platform for predictive maintenance", *Sensors*, Vol. 21 No. 14, p. 4676.

Rieger, T., Regier, S., Stengel, I. and Clarke, N.L. (2019), "Fast predictive maintenance in industrial internet of things (iiot) with deep learning (dl): a review", in *CERC*, pp. 69-80.

Sala, R., Bertoni, M., Pirola, F. and Pezzotta, G. (2021), "Data-based decision-making in maintenance service delivery: the d3m framework", *Journal of Manufacturing Technology Management*, Vol. 32 No. 9, pp. 122-141.

Saxena, A., Goebel, K., Simon, D. and Eklund, N. (2008), "Damage propagation modeling for aircraft engine run-to-failure simulation", *2008 International Conference on Prognostics and Health Management*, Denver, CO, USA, pp. 1-9, doi: 10.1109/PHM.2008.4711414.

Sharma, B., Sharma, L. and Lal, C. (2019), "Anomaly detection techniques using deep learning in iot: a survey", *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 146-149.

Shcherbakov, M. and Sai, C. (2022), "A hybrid deep learning framework for intelligent predictive maintenance of cyber-physical systems", *ACM Transactions on Cyber-Physical Systems*, Vol. 6 No. 2, 17, doi: 10.1145/3486252.

Sohaidan, F.N.B., Muneer, A. and Taib, S.M. (2021), "Remaining useful life prediction of turbofan engine using long-short term memory", *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, IEEE, pp. 1-6.

Song, H., Rajan, D., Thiagarajan, J. and Spanias, A. (2018), "Attend and diagnose: clinical time series analysis using attention models", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32 No. 1.

Susto, G.A., Schirru, A., Pampuri, S., McLoone, S. and Beghi, A. (2015), "Machine learning for predictive maintenance: a multiple classifier approach", *IEEE Transactions on Industrial Informatics*, Vol. 11 No. 3, pp. 812-820.

Teoh, Y.K., Gill, S.S. and Parlikad, A.K. (2023), "IoT and fog-computing-based predictive maintenance model for effective asset management in Industry 4.0 using machine learning", *IEEE Internet of Things Journal*, Vol. 10 No. 3, pp. 2087-2094, doi: 10.1109/JIOT.2021.3050441.

Tortorella, G., Saurin, T.A., Fogliatto, F.S., Tlapa, D., Moyano-Fuentes, J., Gaiardelli, P., Seyedghorban, Z., Vassolo, R., Mac Cawley, A.F., Sreedharan, V.R., *et al.* (2022), "The impact of industry 4.0 on the relationship between tpm and maintenance performance", *Journal of Manufacturing Technology Management*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017), "Attention is all you need", in Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R. (Eds), *Advances in Neural Information Processing Systems*, Curran Associates, Vol. 30, available at: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wang, Q., Bu, S. and He, Z. (2020), "Achieving predictive and proactive maintenance for high-speed railway power equipment with lstm-rnn", *IEEE Transactions on Industrial Informatics*, Vol. 16 No. 10, pp. 6509-6517.

Wu, H., Huang, A. and Sutherland, J.W. (2020), "Avoiding environmental consequences of equipment failure via an lstm-based model for predictive maintenance", Procedia Manufacturing, *Sustainable Manufacturing - Hand in Hand to Sustainability on Globe: Proceedings of the 17th Global Conference on Sustainable Manufacturing*, Vol. 43, pp. 666-673.

Yang, H., Li, Z., Qiang, H., Li, Z., Tu, Y. and Yang, Y. (2020), "Zte-predictor: disk failure prediction system based on lstm", *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, Valencia, Spain, pp. 17-20, doi: 10.1109/DSN-S50200.2020.00017.

Zare, S. and Ayati, M. (2021), "Simultaneous fault diagnosis of wind turbine using multichannel convolutional neural networks", *ISA Transactions*, Vol. 108, pp. 230-239.

Zhang, M., Amaitik, N., Wang, Z., Xu, Y., Maisuradze, A., Peschl, M. and Tzovaras, D. (2022a), "Predictive maintenance for remanufacturing based on hybrid-driven remaining useful life prediction", *Applied Sciences*, Vol. 12 No. 7, p. 3218.

Zhang, J., Jiang, Y., Wu, S., Li, X., Luo, H. and Yin, S. (2022b), "Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism", *Reliability Engineering andi System Safety*, Vol. 221, 108297, doi: 10.1016/j.ress.2021.108297.

Zhang, W., Yang, D. and Wang, H. (2019), "Data-driven methods for predictive maintenance of industrial equipment: a survey", *IEEE Systems Journal*, Vol. 13 No. 3, pp. 2213-2227.

Corresponding author
Giancarlo Sperlì can be contacted at: giancarlo.sperli@unina.it