Decision Analytics Journal 6 (2023) 100174

Contents lists available at ScienceDirect

Decision Analytics Journal

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

A predictive maintenance model using Long Short-Term Memory Neural Networks and Bayesian inference

Davide Pagano

Department of Mechanical and Industrial Engineering, University of Brescia, Italy

ARTICLE INFO

Keywords:
Predictive maintenance
Long short-term memory
Neural networks
Bayesian inference
Industrial plant

ABSTRACT

The fourth industrial revolution is a profound transformation utilizing emerging technologies like smart automation, large-scale machine-to-machine communication, and the internet of things to change traditional manufacturing and industrial practices. The analysis of the huge amount of data collected in all modern industrial plants not only greatly benefited from modern tools of artificial intelligence but has also spurred the development of new ones. In this context, we present a new approach based on the combined use of Long Short-Term Memory (LSTM) neural networks and Bayesian inference for the predictive maintenance of an industrial plant. Hotelling's T^2 and Q metrics, assessing the degree of compatibility between the time-evolving industrial data and the output of the LSTM, trained on a reference period of good working condition, are used to update the Bayesian posterior probability about the good working condition of the plant. This method has successfully been applied to a real industrial case, and the results are presented and discussed.

1. Introduction

Maintenance plays an increasingly important role in business economics and planning: the goal is to ensure the plant functionality and the cost-effectiveness of its operation, in the sense of reducing production costs and safeguarding all the equipment, which represent a company asset. In the literature (see for example [1,2]), three main maintenance methodologies are identified: (i) *reactive maintenance* (also called *run-to-failure maintenance*); (ii) *preventive maintenance*; (iii) *predictive maintenance*.

In the case of run-to-failure maintenance, maintenance interventions are deferred until a failure occurs in a part of the plant. The purpose of this type of intervention is to restore the operation, and it is mainly employed in cases of non-critical and/or low-cost equipment. Preventive maintenance, on the other hand, is time-based: maintenance operations on the components of a plant occur at predetermined intervals, with the goal of making the plant operate in a regime in which the probability of failure is low. Maintenance interventions are therefore scheduled according to the number of service hours of each component, based on the expected mean life between failures provided by the manufacturer and/or extracted from the literature. Predictive maintenance (PdM) departs from the previously described approaches by aiming at inferring the *health* of a plant, based on the temporal evolution of its condition. The idea is to perform maintenance interventions only when actually needed, resulting in less overall maintenance costs, intended as the sum of those associated to prevention and repair, as shown in Fig. 1.

2. Status-of-the-art and related works

Several techniques have been proposed in literature on how to assess the health of a plant, as measured by means of some metrics, from the analysis of the temporal evolution of its monitoring data. In general, they are classified as physical model-based, knowledge-based, or data-driven [3].

Approaches based on physical models rely on the knowledge of the degradation processes occurring for the individual components of the plant, integrated into typically very complex models, which are then used to simulate the effects of different types of failures on condition data [4–6]. Because of the need for assumptions to define the physical models, this approach to predictive maintenance is typically limited to small parts of the plant or specific components, and this does not allow general conclusions to be drawn about the health of an entire plant.

Knowledge-based approaches try to reduce the complexity of a physical models by taking advantage of the current knowledge of a system, for example through a set of rules defined by experts. Approaches like this are sometimes employed in large industrial plants, where there is a lot of experience about its behavior and typical failures, but there is not enough knowledge to develop a quantitative models [7].

More relevant for this work are, finally, the data-driven techniques, which only rely on collected monitoring data and maintenance events history. They take advantage of the high flexibility of Machine Learning (ML) techniques to learn how the plant behaves when working

E-mail address: davide.pagano@unibs.it.

https://doi.org/10.1016/j.dajour.2023.100174
Received 1 December 2022; Received in revised form 13 January 2023; Accepted 23 January 2023
Available online 25 January 2023

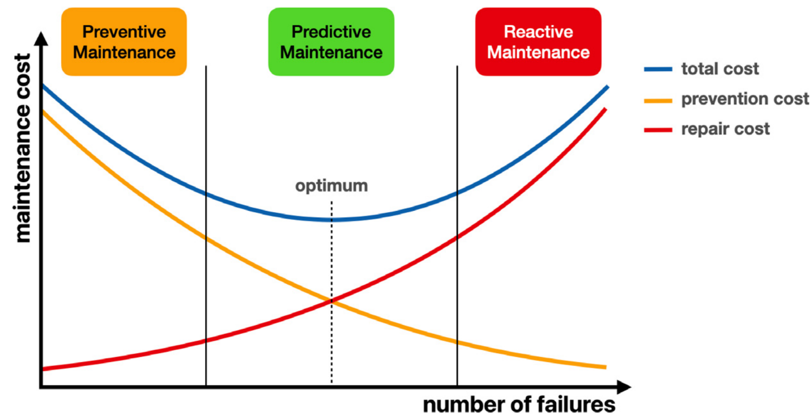

Fig. 1. Maintenance costs, as intended as the sum of those associated to prevention and repair, as a function of the number of failures.

in optimal conditions, as well as in presence of failures. The learning by ML algorithms can follow three main approaches: supervised, unsupervised, and semi-supervised.

Supervised learning, in addition to data, also requires the knowledge about the system's performance at the time data refer to. Data are therefore labeled according to several categories associated to different working conditions of the plant, with and without failures. The PdM problem can be dealt with as a classification task or, if data can be labeled in a continuous way, as a regression problem [8]. In this context, a plethora of ML techniques have been investigated over the years, as summarized in Table 1, which also presents a selection of the most recent literature for each ML method. Of course the list is far from comprehensive, as it goes beyond the scope of this manuscript, so we refer the reader to the following recent literature reviews [7–10] for more details.

Unsupervised learning aims at finding patterns in data and relationships between their features. It does not require any labeling of data and, in the context of PdM, is typically used to identify different working conditions of a system by means of clustering tasks. A hybrid paradigm, between supervised and unsupervised learning, is the semi-supervised approach, which works with both labeled and unlabeled data. In the context of PdM, this approach could be useful when a sufficiently large record of failures of a system is missing, like in the industrial case presented in this work. In this case, the underlying idea is to make predictions on the basis of models, trained with reference (labeled) data, in order to understand the time evolution of data and to assess, from the comparison with the actual measurements, the presence of degradation trends or other features which could be traced back to a problem in the plant. Table 1 also presents a (non-comprehensive) selection of the most recent literature for unsupervised and semi-supervised techniques used in the context of PdM.

In this manuscript a technique based on the combined use of long short-term memory networks (LSTM) and Bayesian inference for the assessment of the health of an industrial plant, in the context of predictive maintenance, is presented. This approach has been successfully applied to an industrial plant and the results are also presented. It is important to highlight, though, that because of a non-disclosure agreement, no detail about the company, as well as their data, will be released. Even though the data presented in the following are not sensitive, as condition data of some active parts of the plant, variable names, units of measurement and scales will be omitted in all plots, not to allow their identification. Of course, the company has given its consent to the publication of the plots in the form in which they are presented in this manuscript.

Table 1
A selection of the most recent literature on ML-based data driven approaches to PdM.

Learning	ML technique	References
Supervised	Decision Trees	Arena et al. (2022) [11] Ouadah et al. (2022) [12] Dangut et al. (2021) [13] Cakir et al. (2021) [14]
	Support Vector Machines	Nikfar et al. (2022) [15] Baptista et al. (2021) [16] Aremu et al. (2021) [17] Lin (2021) [18]
	Neural Networks	Zonta et al. (2022) [19] Demidova (2021) [20] Lv et al. (2021) [21] Sun et al. (2021) [22] Morariu et al. (2020) [23] Silva et al. (2019) [24]
	Linear/Logistic Regression	Kang et al. (2021) [25] Saranya et al. (2020) [26] Keartland et al. (2020) [27] Zenisek et al. (2019) [28]
	K-Nearest Neighbors	Xu et al. (2021) [29] Pinto et al. (2019) [30]
Unsupervised and Semi-supervised	Partitional clustering	Giordano et al. (2021) [31] Kamat et al. (2021) [32] Bekar et al. (2020) [33] Oliveira et al. (2020) [34]
	Hierarchical and Density-based Clustering	Serradilla et al. (2021) [35] Aremu et al. (2020) [36] Hsu et al. (2020) [37] Zschech et al. (2019) [38]
	Deep Learning	Wang et al. (2022) [39] Malawade et al. (2021) [40] Basora et al. (2021) [41] Ning et al . (2021) [42] Cho et al. (2020) [43]

3. The method

The basic idea of the proposed method, that is to model the condition data in order to study how the agreement between model predictions and data evolve over time, is common to many data-driven techniques for predictive maintenance already available in literature. Indeed, differences between the various approaches rely on the way the modeling is performed, on how to measure the agreement between data and predictions, and on how to use this information to assess the overall health of the plant.

This work was triggered by a real industrial case: a company (in the heavy industry) wanted to investigate the possibility of implementing a PdM program, on the basis of the information from the monitoring sensors already installed in their plant. From a preliminary analysis of the available data, we identified two constraints to take into account: (i) the lack of a proper record of failures for each active/moving part of the plant; (ii) the availability of only time series, with different levels of autocorrelation among the variables, as input data. The first

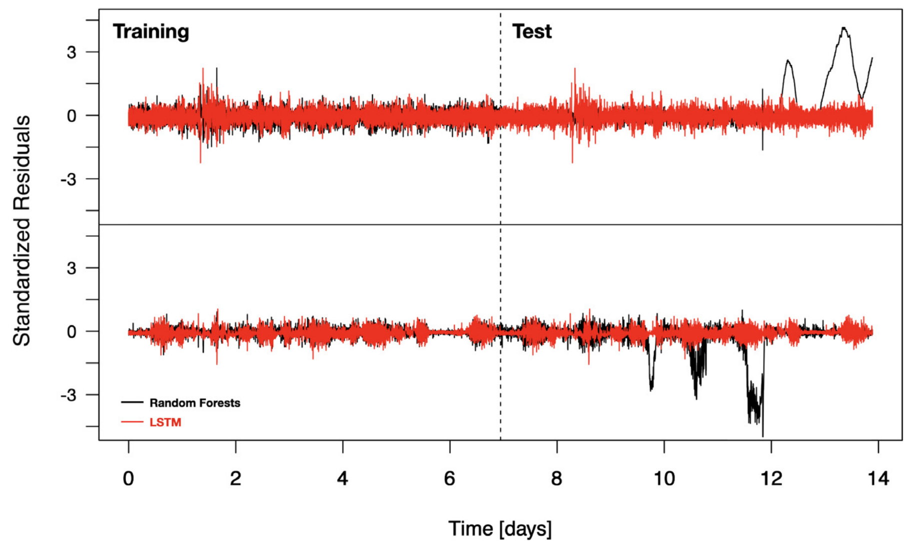

Fig. 2. Standardized residuals for two variables from an active part of the plant, as obtained from the use of a (multivariate) Random Forests approach. The vertical dashed line separates the training period from the test one.

point basically prevented the use of supervised techniques, whereas the second one posed some troubles to other popular approaches available in literature, as described below.

It is well know that a time series can be framed as a supervised regression problem with techniques such as the sliding-window representation [44–47]. Thanks to this, several popular ML techniques can be employed to times series and, in particular, we focused on multivariate approaches in order to exploit the possible correlations among variables. Preliminary tests showed, thought, that some of the most popular approaches in literature, such as Random Forests and Deep Learning, tended to struggle in predicting the trends of several monitoring variables of the plant, especially on a time scale longer than 1 or 2 weeks. As an example, Fig. 2 shows the standardized residuals (residuals divided by their standard deviation) for two variables associated to an engine of the plant. Both training and test data (separated by the dashed line in the plot) belong to a period of known good working conditions of the part. The black curve is what was obtained with a (multivariate) Random Forests [48] (RF), wheres the red line is what was obtained with a Long-Short Memory Network [49], investigated in this work.

As it can be noticed, while the performance of the two methods in the training period are similar, the results are very different in the test period. In particular, a set of peaks, associated to large discrepancies between predictions and data, are present for the RF but not for the LSTM. Those peaks appear in both variables but at very different times and cannot be associated to any issue with the engine.

As already mentioned, we ended up investigating the use of long-short memory networks [49] to model the condition data of the industrial plant we analyzed. The LSTM architecture has been proposed to solve the well known *vanishing gradient problem* [50] with recurrent neural networks. Nowadays, the most commonly used setup for LSTM networks is the one proposed by Graves and Schmidhuber [51], generally referred to as *vanilla LSTM*, which is also the one used in this work. It consists of a set of recurrently connected memory blocks, each containing one or more self-connected memory cells and three multiplicative units: input, output and forget gates, the latter missing in the original formulation of Hochreiter [49]. The memory cells remember values over arbitrary time intervals and the three gates regulate the flow of information through the cell. While over the years several other variations of the LSTM architecture have been proposed, the results of a large-scale study conducted by Gref et al. [52] showed that none of the eight investigated LSTM variations significantly improved the performance of the vanilla LSTM.

We decided to model each active component of the plant (such as fans, engines, etc.) with LSTM networks, as very well-suited to make predictions based on time series data. In general, even restricting their use to individual active units, it is very likely that the number of variables to model is high and that some of them are not very informative. This was indeed the scenario with the industrial case we analyzed. For this reason, after having dealt with the presence of fully correlated variables, of missing values and placeholder values, and of variables with zero or near-zero variance, we performed a *principal components analysis*, which is a orthogonal linear transformation of the variables into new ones in a way where most of the variation in data can be described with fewer dimensions [53].

For a $m \times n$ data matrix $\boldsymbol{X}$, the PCA model is defined by the sum of k outer products of vectors $\boldsymbol{t}_i$ and $\boldsymbol{p}_i$, plus a residual matrix $\boldsymbol{E}$, describing the unmodeled information (residual variance [54]:

$$\boldsymbol{X} = \boldsymbol{t}_1\boldsymbol{p}_1^T + \boldsymbol{t}_2\boldsymbol{p}_2^T + \cdots + \boldsymbol{t}_k\boldsymbol{p}_k^T + \boldsymbol{E} = \boldsymbol{T}_k\boldsymbol{P}_k^T + \boldsymbol{E}.$$

Vectors $\boldsymbol{t}_i$ are known as principal component *scores* and are a linear combination of the original data $\boldsymbol{X}$, through the transformation vectors $\boldsymbol{p}_i$, which are the eigenvectors of the covariance of matrix $\boldsymbol{X}$: $\boldsymbol{X}\boldsymbol{p}_i = \boldsymbol{t}_i$. The eigenvalues associated with each eigenvector $\boldsymbol{p}_i$ provide the information on how much variance in data is explained by each principal component.

In our industrial case, depending on the active component, we cut on the cumulative variance between 85% and 90%, corresponding to a number of principal components between 20 and 40. Thanks to this, at the cost of small loss in the variance explained, we reduced the number of variables up to 55%. The networks were then trained on the resulting PCA models. As an example, Fig. 3 shows the cumulative variance as a function of the number of principal components (only the first 30 are reported), for a particular active unit (an engine) of the plant.

For several active parts of the plant, we used a vanilla LSTM network, consisting of a number of input and output units equal to the number n of features, and a number of hidden units between 32 and 128 (depending on n). In the other cases, instead, because of the very large number of variables we got better performance with a stacked LSTM network, that is, with multiple hidden LSTM layers stacked one on top of another. We used a configuration with two hidden layers and a number of units between 32 and 64 per layer. In all cases the training was performed using the *mean squared error* as loss function (although tests with other regression losses gave very similar results) and the *Adam* optimizer [55], which is a first-order gradient-based optimization algorithm, based on adaptive estimates of lower-order moments.

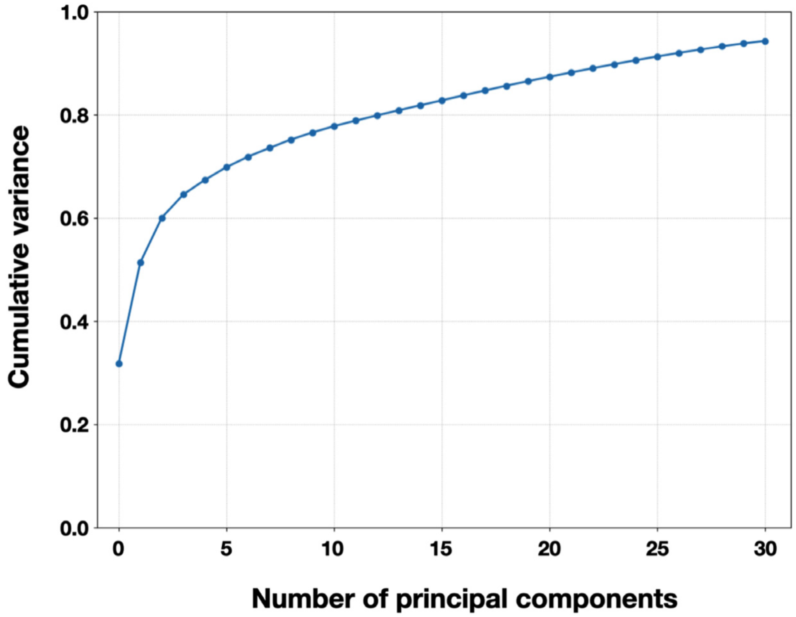

Fig. 3. Cumulative variance as a function of the number of principal components (only the first 30 are reported), for a PCA performed on an active unit of the plant.

Before training the networks we also performed a re-sampling of the variables, as for some time series the sampling frequency was too high in relation to the measured physical phenomenon. This was indeed the case of many temperature sensors, sampled up to 100 Hz. After having prepared the data, we modeled them during a period when the plant was in a supposedly good working conditions. More specifically, we identified, together with the company, temporal windows, generally following operations of preventive maintenance, where there was high belief about the good health of the plant. The training of the networks was performed with data in those temporal regions. As working conditions, especially for some components, can change quickly, training data spanned over 15 days at most.

The first step was to check whether the trained models could be used to detect anomalies in data, which could be associated to a change in the working conditions of the plant. Fig. 4 shows 4 variables (out of 28) associated to an active component of the plant we analyzed. The first column of plots shows the 15 days of data used as training period, whereas the second one shows the following 45 days, used as test period.

The initial agreement in the test region, between model and data, was very good for all variables until the barred area, starting at day 11, when some tension started to appear in most of the variables. Indeed, during the period between day 11 and day 18, there was a shutdown of that active part and a maintenance intervention on it and, therefore, afterwards it started working in different conditions with respect to those used for the training. The sensitivity of this approach to changes in the working conditions of the plant was confirmed by several other tests, by using test data including maintenance interventions.

4. Early detection of failures

A wide range of failures, which can occur in an industrial plant, are preceded by the appearance of trends or other changes in the condition data, which could be very difficult to notice, especially at the onset of the problem. For this reasons, with the help of the company, we analyzed a set of historical data, where there may or may not have been a problem with an active component of the plant. We did not know this information when analyzing the data. In the following, the results of one those tests are presented.

Fig. 5 shows some of the data, associated to an active component of the plant, we used for the training and test, separated by the superimposed red line. For visualization clarity, only 6 variables (out of 29) are reported in plot and also the test period has been limited to 15 days (out of 45 days).

In order to measure the agreement between predictions from the model and data, we used both the Hotteling's T^2 and Q metrics, which are common statistics in evaluating new (test) data using models based on PCAs. The T^2-statistic is usually thought as a measure of the variation within the model and is defined as the sum of the normalized squared scores: $T_i^2 = \boldsymbol{t}_i(\boldsymbol{T}_k^T\boldsymbol{T}_k)^{-1}\boldsymbol{t}_i^T$. The Q-statistic, instead, is usually considered as a measure of the unexplained variance in the model and is defined as the difference between the test data point and its projection on the PCA model [56]. Fig. 6 shows the agreement between data and predictions of 3 variables and for the first 15 days of the test period.

Initially, the observed agreement was very good, but, at approximately the time indicated by the dashed line, we starting seeing an increasing tension, between data and predictions, as measured by T^2 and Q metrics. Variables shown in Fig. 6 are those for which the increasing tensions was more visible and, indeed, it can be also appreciated by eye at some point.

The dashed line in Fig. 6 corresponds to one in Fig. 7, which shows the measured values for T^2 and Q for a subset of the test period. As it can be noted, values of both metrics really exploded in a time scale of the order of 1 h, starting approximately at 11 pm. The reason for this increasing mismatch, between data and predictions, was actually due a failure in the active components under analysis, which, at the time when these data refer to, was identified by the company at some point. The interesting thing is that the time at which we started observing an increase of the values for both metrics anticipates by many hours the time at which the problem was identified by the company: we actually performed an early detection of a failure, also confirmed with similar other tests.

5. Assessment of the health of the plant

Having proved that this technique was sensitive to changes in the working condition of the plant, we then tried to use the information from T^2 and Q metrics to infer the actual health of the plant. We did this in a Bayesian fashion.

Let $\boldsymbol{\omega}$ be a set of possible states describing the health of a plant and $\boldsymbol{x} = \{T^2, Q\}$ a measurement of the agreement between the model

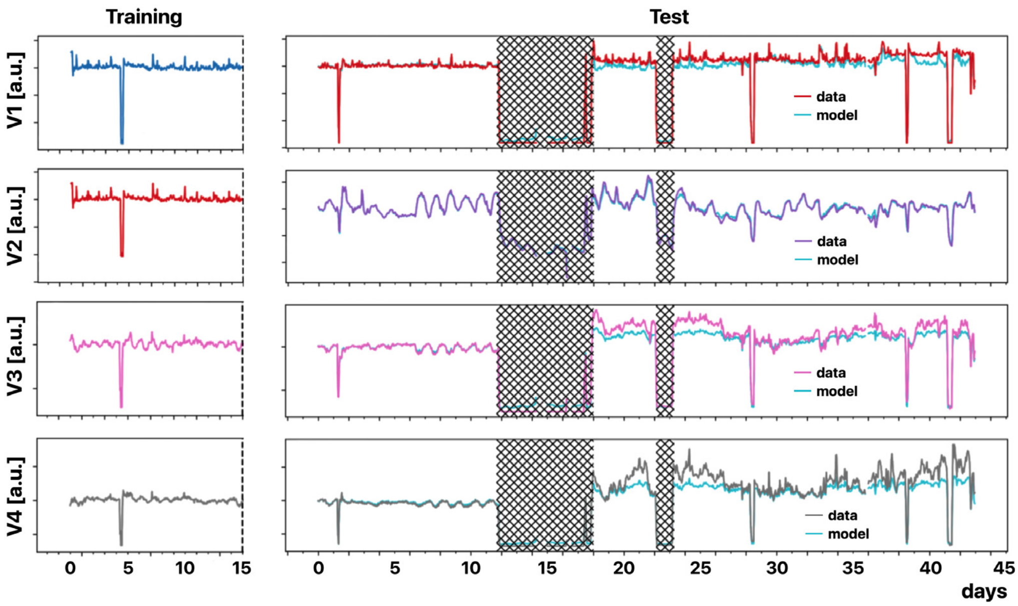

Fig. 4. Comparison between data and LSTM predictions for one active component of the plant. Left: training period; Right: test period. Only four variables are reported for visualization clarity.

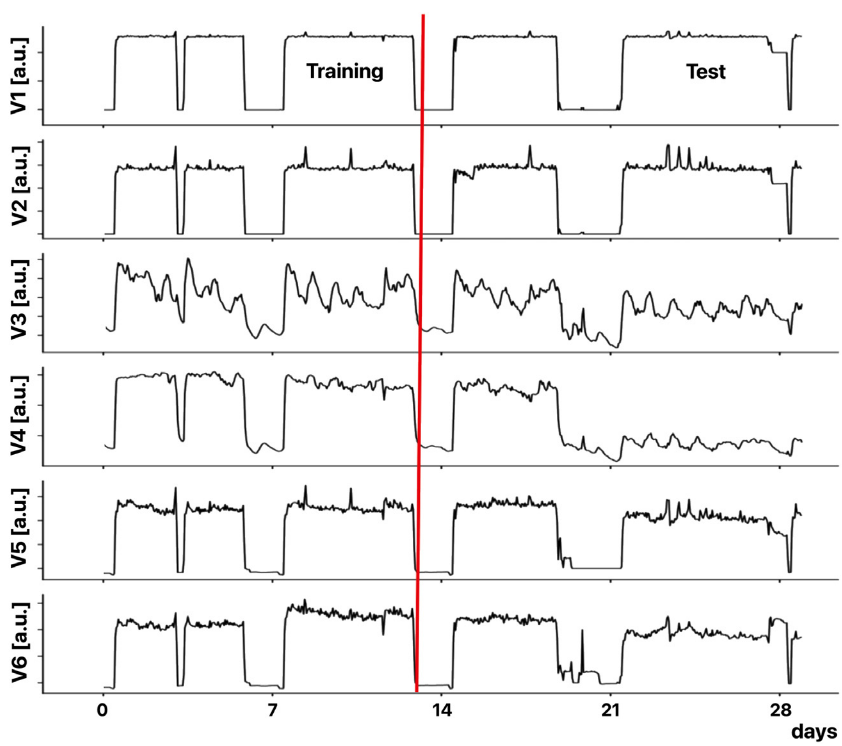

Fig. 5. Training and test data, as defined by the superimposed red line, used for the blind search of anomalies, as described in the text. Only six variables are reported for visualization clarity.

and data. According to the Bayes' theorem, the posterior probability $P(\omega_j|\boldsymbol{x})$, for each $\omega_j \in \boldsymbol{\omega}$, is given by Eq. (1),

$$P(\omega_j|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_j)P(\omega_j)}{\int p(\boldsymbol{x}|\omega_j)P(\omega_j)d\omega_j} \tag{1}$$

where $p(\boldsymbol{x}|\omega_j)$ is the likelihood of $\boldsymbol{x}$ given ω_j, $P(\omega_j)$ is the prior probability of ω_j, and the denominator is the marginal probability of $\boldsymbol{x}$.

In the most simple approach one can simply define two possible states associated to the health of a plant: $\boldsymbol{\omega} = \{\omega_g, \omega_b\}$, where ω_g (ω_b) describes the scenario of a plant in a good (bad) working condition. The likelihoods $p(\boldsymbol{x}|\omega_j)$ were estimated from data (at least as far as $p(\boldsymbol{x}|\omega_g)$ is concerned): the joint pdf $p(\boldsymbol{x}|\omega_g)$ was modeled as the product of two Normal distributions, whereas uniform distributions were used for $p(\boldsymbol{x}|\omega_b)$.

It important to highlight that $p(\boldsymbol{x}|\omega_b)$ depends on the type of problem experienced by the plant and, therefore, it cannot be easily modeled from data, especially in absence of long record of failures, which (luckily for the company) was the case. So, the choice of uniform distributions was a sort of compromise between simplicity and performance,

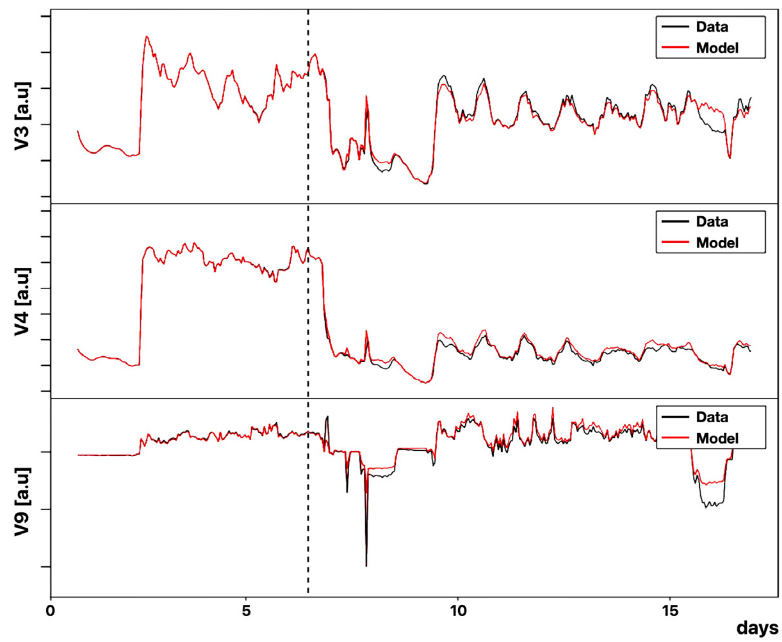

Fig. 6. Comparison between model predictions and data, in the test period, for 3 variables of the analyzed active component of the plant. The dashed line indicates when we started observing an increasing tension, as measured by T^2 and Q metrics.

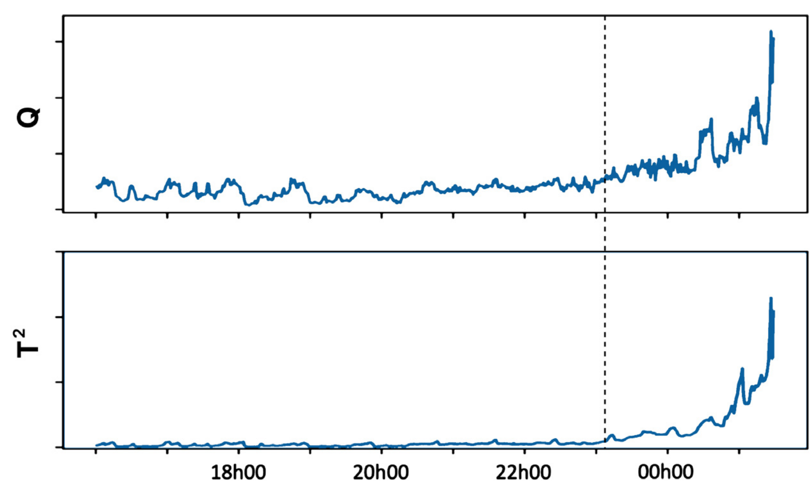

Fig. 7. Hotteling's T^2 and Q metrics as a function of the time, for a sub-range of the test data in Fig. 6. Dashed lines in the two figures correspond to the same time.

as a result of several tests. In principle $\boldsymbol{\omega}$ can include different possible states describing bad working conditions, for example, by specifying the type of failure. In this way, not only a more refined modeling of $p(\boldsymbol{x}|\omega_j)$ is possible, but the proposed method could also suggest the type of problem the plant may be experiencing.

Fig. 8 shows the posterior probability $p(\boldsymbol{x}|\omega_g)$ as a function of the time for the test data in Fig. 5, in a short temporal window around the time at which we started observing an increase of T^2 and Q metrics, that is around the dashed line in Figs. 6 and 7. Initially, when the agreement between data and model was still quite good, the posterior probability remains very high, close to the initial prior probability decided with the company, on the basis of the full inspection of the plant. But then, when this good agreement ends, the posterior probability starts decreasing and in a time windows of very few hours drops to very low values. This posterior probability can be seen as a measurement of the good health of the plant and this value can trigger an alarm, as illustrated in the flowchart in Fig. 9.

6. Conclusions

This manuscript presented the use of LSTM networks and Bayesian inference in the context of predictive maintenance. The proposed method consists of two steps: (i) monitoring data associated to each active component of the plant are modeled, after a PCA to reduce their dimension, with these type of neural networks, by using appropriately chosen training periods, where the plant is in supposedly good working conditions; (ii) Hotteling's T^2 and Q metrics are used to measure the degree of agreement between network predictions and test data, and their values update the posterior probability of the good health of the component, given the data, in a Bayesian fashion.

This method has been successfully applied to an industrial case. Despite having only considered two possible states for $\boldsymbol{\omega}$ and having simply used uniform distributions for $p(\boldsymbol{x}|\omega_b)$, we obtained good results from many tests we performed. Because of this, the presented technique has been integrated in a specifically designed dashboard, which is now

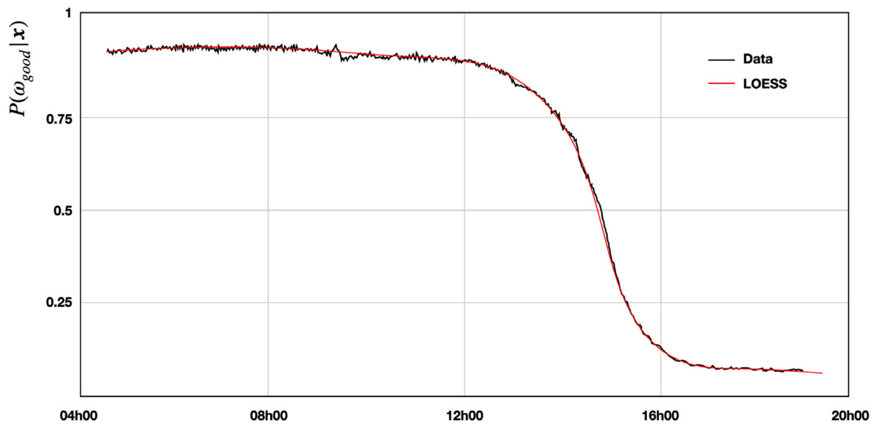

Fig. 8. Posterior probability $P(\omega_j|\mathbf{x})$ of the good health of the component, given the data, as a function of data. The presented temporal window is around the dashed black lines in Fig. 7 and 6, as described in the text.

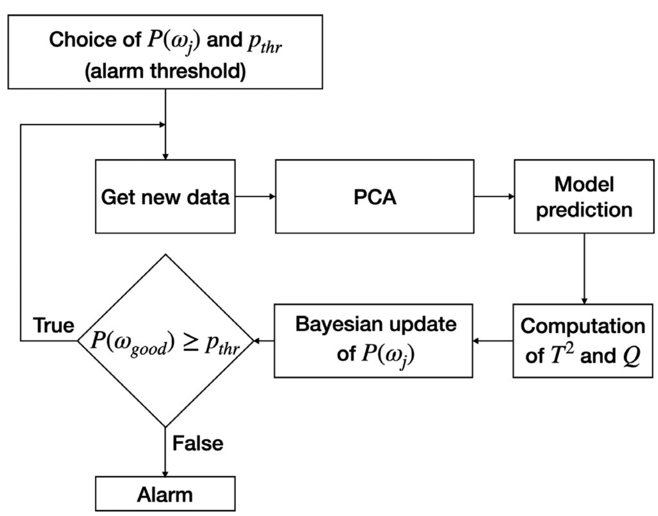

Fig. 9. Flowchart of the proposed method. After the training phase (not represented in the Figure), the user chooses the initial priors $P(\omega_j)$ and the alarm threshold p_{thr}. The PCA is performed for all new data, to reduce their dimension, and then the model prediction for them is computed. Hotteling's T^2 and Q metrics are calculated and used to update $P(\omega_j)$ according to the Bayes' theorem. As long as $P(\omega_{good}) \geq p_{thr}$ the procedure is iterated, otherwise an alarm is raised.

part of the tools used by the company for the maintenance of their plant.

As a final note, while this work was triggered and focused on a particular study case, that is the predictive maintenance of an industrial plant, it is worth to mention that the proposed technique can extended to many other scenarios, where time-evolving systems are involved. It is important to highlight, though, two possible limitations related to the method. The first one, obvious and common to all Bayesian methods, is the possible lack of sensitivity if the likelihoods $p(\mathbf{x}|\omega_j)$ are very similar to each other. The other, more subtle, is the possible appearance of large fluctuations in the posterior probability because of noisy data. Indeed, in this multivariate context, occasionally large values of the T^2 and Q metrics could be just the result of random fluctuations of data because of noise. As a result, a large variation in the posterior probability could be induced. While this effect was not an issue for the analyzed data, also thanks to the re-sampling of the variables reducing their noise level, it could be an issue for other cases. However, these large fluctuations of the posterior probability can be regularized, for example by performing the Bayesian update on the averaged values of the metrics over batches of data.

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

The author acknowledges the support from the University of Brescia, Italy and the company which has funded this research.

References

[1] R.K. Mobley, 1 - impact of maintenance, in: R.K. Mobley (Ed.), An Introduction To Predictive Maintenance (Second Edition), second ed., in: Plant Engineering, Butterworth-Heinemann, Burlington, 2002, pp. 1–22, http://dx.doi.org/10.1016/B978-075067531-4/50001-4.

[2] J. Levitt, Complete Guide To Predictive and Preventive Maintenance, second ed., Industrial Press, New York, NY, 2011.

[3] T. Zonta, C.A. Da Costa, R. da Rosa Righi, M.J. de Lima, E.S. da Trindade, G.P. Li, Predictive maintenance in the industry 4.0: A systematic literature review, Comput. Ind. Eng. 150 (2020) 106889.

[4] A.K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mech. Syst. Signal Process. 20 (7) (2006) 1483–1510.

[5] Y. Li, S. Billington, C. Zhang, T. Kurfess, S. Danyluk, S. Liang, Adaptive prognostics for rolling element bearing condition, Mech. Syst. Signal Process. 13 (1) (1999) 103–113.

[6] C.J. Li, H. Lee, Gear fatigue crack prognosis using embedded model, gear dynamic model and fracture mechanics, Mech. Syst. Signal Process. 19 (4) (2005) 836–846.

[7] Y. Ran, X. Zhou, P. Lin, Y. Wen, R. Deng, A survey of predictive maintenance: Systems, purposes and approaches, 2019, arXiv preprint arXiv:1912.07383.

[8] A. Esteban, A. Zafra, S. Ventura, Data mining in predictive maintenance systems: A taxonomy and systematic review, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 12 (5) (2022) e1471.

[9] O. Serradilla, E. Zugasti, J. Rodriguez, U. Zurutuza, Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects, Appl. Intell. (2022) 1–31.

[10] T.P. Carvalho, F.A. Soares, R. Vita, R.d.P. Francisco, J.P. Basto, S.G. Alcalá, A systematic literature review of machine learning methods applied to predictive maintenance, Comput. Ind. Eng. 137 (2019) 106024.

[11] S. Arena, E. Florian, I. Zennaro, P. Orrù, F. Sgarbossa, A novel decision support system for managing predictive maintenance strategies based on machine learning approaches, Saf. Sci. 146 (2022) 105529.

[12] A. Ouadah, L. Zemmouchi-Ghomari, N. Salhi, Selecting an appropriate supervised machine learning algorithm for predictive maintenance, Int. J. Adv. Manuf. Technol. 119 (7) (2022) 4277–4301.

[13] M.D. Dangut, Z. Skaf, I.K. Jennions, An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset, ISA Trans. 113 (2021) 127–139.
[14] M. Cakir, M.A. Guvenc, S. Mistikoglu, The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system, Comput. Ind. Eng. 151 (2021) 106948.
[15] M. Nikfar, J. Bitencourt, K. Mykoniatis, A two-phase machine learning approach for predictive maintenance of low voltage industrial motors, Procedia Comput. Sci. 200 (2022) 111–120.
[16] M.L. Baptista, E.M. Henriques, H. Prendinger, Classification prognostics approaches in aviation, Measurement 182 (2021) 109756.
[17] O.O. Aremu, D. Hyland-Wood, P.R. McAree, A machine learning approach to circumventing the curse of dimensionality in discontinuous time series machine data, Reliab. Eng. Syst. Saf. 195 (2020) 106706.
[18] S.-L. Lin, Application of machine learning to a medium Gaussian support vector machine in the diagnosis of motor bearing faults, Electronics 10 (18) (2021) 2266.
[19] T. Zonta, C.A. da Costa, F.A. Zeiser, G. de Oliveira Ramos, R. Kunst, R. da Rosa Righi, A predictive maintenance model for optimizing production schedule using deep neural networks, J. Manuf. Syst. 62 (2022) 450–462.
[20] L. Demidova, Recurrent neural networks' configurations in the predictive maintenance problems, IOP Conf. Ser. Mater. Sci. Eng. 714 (1) (2020) 012005.
[21] Y. Lv, Q. Zhou, Y. Li, W. Li, A predictive maintenance system for multi-granularity faults based on AdaBelief-BP neural network and fuzzy decision making, Adv. Eng. Inform. 49 (2021) 101318.
[22] L. Sun, T. Liu, Y. Xie, D. Zhang, X. Xia, Real-time power prediction approach for turbine using deep learning techniques, Energy 233 (2021) 121130.
[23] C. Morariu, O. Morariu, S. Răileanu, T. Borangiu, Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems, Comput. Ind. 120 (2020) 103244.
[24] W. Silva, M. Capretz, Assets predictive maintenance using convolutional neural networks, in: 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD, IEEE, 2019, pp. 59–66.
[25] Z. Kang, C. Catal, B. Tekinerdogan, Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks, Sensors 21 (3) (2021) 932.
[26] E. Saranya, P.B. Sivakumar, Data-driven prognostics for run-to-failure data employing machine learning models, in: 2020 International Conference on Inventive Computation Technologies, ICICT, IEEE, 2020, pp. 528–533.
[27] S. Keartland, T.L. Van Zyl, Automating predictive maintenance using oil analysis and machine learning, in: 2020 International SAUPEC/RobMech/PRASA Conference, IEEE, 2020, pp. 1–6.
[28] J. Zenisek, F. Holzinger, M. Affenzeller, Machine learning based concept drift detection for predictive maintenance, Comput. Ind. Eng. 137 (2019) 106031.
[29] Y. Xu, N.M.M. Nascimento, P.H.F. de Sousa, F.G. Nogueira, B.C. Torrico, T. Han, C. Jia, P.P. Rebouças Filho, Multi-sensor edge computing architecture for identification of failures short-circuits in wind turbine generators, Appl. Soft Comput. 101 (2021) 107053.
[30] R. Pinto, T. Cerquitelli, Robot fault detection and remaining life estimation for predictive maintenance, Procedia Comput. Sci. 151 (2019) 709–716.
[31] D. Giordano, M. Mellia, T. Cerquitelli, K-mdtsc: K-multi-dimensional time-series clustering algorithm, Electronics 10 (10) (2021) 1166.
[32] P.V. Kamat, R. Sugandhi, S. Kumar, Deep learning-based anomaly-onset aware remaining useful life estimation of bearings, PeerJ Comput. Sci. 7 (2021) e795.
[33] E.T. Bekar, P. Nyqvist, A. Skoogh, An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study, Adv. Mech. Eng. 12 (5) (2020) 1687814020919207.
[34] M.A. Oliveira, E.F. Simas Filho, M.C. Albuquerque, Y.T. Santos, I.C. Da Silva, C.T. Farias, Ultrasound-based identification of damage in wind turbine blades using novelty detection, Ultrasonics 108 (2020) 106166.
[35] O. Serradilla, E. Zugasti, J. Ramirez de Okariz, J. Rodriguez, U. Zurutuza, Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data, Appl. Sci. 11 (16) (2021) 7376.
[36] O.O. Aremu, R.A. Cody, D. Hyland-Wood, P.R. McAree, A relative entropy based feature selection framework for asset data in predictive maintenance, Comput. Ind. Eng. 145 (2020) 106536.
[37] J.-Y. Hsu, Y.-F. Wang, K.-C. Lin, M.-Y. Chen, J.H.-Y. Hsu, Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning, Ieee Access 8 (2020) 23427–23439.
[38] P. Zschech, K. Heinrich, R. Bink, J.S. Neufeld, Prognostic model development with missing labels, Bus. Inform. Syst. Eng. 61 (3) (2019) 327–343.
[39] H. Wang, W. Zhang, D. Yang, Y. Xiang, Deep-learning-enabled predictive maintenance in industrial internet of things: Methods, applications, and challenges, IEEE Syst. J. (2022).
[40] A.V. Malawade, N.D. Costa, D. Muthirayan, P.P. Khargonekar, M.A. Al Faruque, Neuroscience-inspired algorithms for the predictive maintenance of manufacturing systems, IEEE Trans. Ind. Inform. 17 (12) (2021) 7980–7990.
[41] L. Basora, P. Bry, X. Olive, F. Freeman, Aircraft fleet health monitoring with anomaly detection techniques, Aerospace 8 (4) (2021) 103.
[42] S. Ning, J. Sun, C. Liu, Y. Yi, Applications of deep learning in big data analytics for aircraft complex system anomaly detection, Proc. Inst. Mech. Eng. O 235 (5) (2021) 923–940.
[43] A.D. Cho, R.A. Carrasco, G.A. Ruz, J.L. Ortiz, Slow degradation fault detection in a harsh environment, IEEE Access 8 (2020) 175904–175920.
[44] A.L. Mohamad Hanapi, M. Othman, R. Sokkalingam, N. Ramli, A. Husin, P. Vasant, A novel fuzzy linear regression sliding window garch model for time-series forecasting, Appl. Sci. 10 (6) (2020) 1949.
[45] I.A. Khan, A. Akber, Y. Xu, Sliding window regression based short-term load forecasting of a multi-area power system, in: 2019 IEEE Canadian Conference of Electrical and Computer Engineering, CCECE, IEEE, 2019, pp. 1–5.
[46] H. Hota, R. Handa, A. Shrivas, Time series data prediction using sliding window based RBF neural network, Int. J. Comput. Intell. Res. 13 (5) (2017) 1145–1156.
[47] M. Vafaeipour, O. Rahbari, M.A. Rosen, F. Fazelpour, P. Ansarirad, Application of sliding window technique for prediction of wind velocity time series, Int. J. Energy Environ. Eng. 5 (2) (2014) 1–7.
[48] M. Segal, Y. Xiao, Multivariate random forests, Wiley Interdiscip. Rev. Data Min. Knowl. Discovery 1 (1) (2011) 80–87.
[49] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[50] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, A field guide to dynamical recurrent neural networks. IEEE Press In, 2001.
[51] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (5–6) (2005) 602–610.
[52] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2222–2232.
[53] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Phil. Trans. R. Soc. A 374 (2065) (2016) 20150202.
[54] N.B. Gallagher, B.M. Wise, S.W. Butler, D.D. White, G.G. Barna, Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process: Improving robustness through model updating, IFAC Proc. Vol. 30 (9) (1997) 79–84, http://dx.doi.org/10.1016/S1474-6670(17)43143-0, IFAC Symposium on Advanced Control of Chemical Processes 1997 (ADCHEM '97), Banff, Canada, 9-11 June.
[55] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[56] R. Penha, J.W. Hines, Using principal component analysis modeling to monitor temperature sensors in a nuclear research reactor, in: Maintenance and Reliability Conference (MARCON 2001), 2001.