Reliability Engineering and System Safety 237 (2023) 109332

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

A two-stage data-driven approach to remaining useful life prediction via long short-term memory networks

Huixin Zhang [a], Xiaopeng Xi [a,*], Rong Pan [b]

[a] College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China
[b] School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, USA

ARTICLE INFO

Keywords:
Remaining useful life
Prognostic
Health indicator
Long short-term memory
Time delay neural network

ABSTRACT

Accurate remaining useful life (RUL) prediction is of great importance for predictive maintenance. With the recent advancements in sensor technology and artificial intelligence, the data-driven approaches to RUL prediction of industrial equipment have gained a lot of attention. However, past researches have not adequately considered the variety of degradation rates and the accumulated information in degradation processes. To deal with this problem, a novel two-stage machine learning approach of RUL prediction is proposed in this paper. A set of nonlinear health indicator functions are constructed to guide the training process of a long short-term memory learner of degradation processes, then a time delay neural network is utilized for RUL prediction. The superiority of the proposed approach in terms of prediction accuracy and conservativeness is demonstrated by a case study of rolling element bearing dataset.

1. Introduction

Modern industries regard the accurate equipment remaining useful life (RUL) prediction technology as a critical asset for ensuring operation safety, equipment availability, and maintenance cost reduction [1–6]. In general, there are two groups of RUL prediction methodologies, i.e., the model-based prediction methodology and the data-driven prediction methodology [7,8]. Model-based methods investigate physical failure mechanisms to establish the degradation process model of machinery. Obtaining the physical failure mechanism of a high-reliability machinery is labor-intensive and time-consuming due to the complexity and increasing durability of equipment nowadays. Obtaining an accurate failure physics-grounded degradation model, although highly desirable, is often illusive in practice. In recent years, data-driven methods, which rely on the powerful pattern recognition capabilities of machine learning (ML) algorithms to extract hidden information from equipment sensor data, have been showing promising prospects in the field of prognostics and health management (PHM).

ML technologies, including support vector machine (SVM) [9,10], random forest [11], neural networks [12] and ensemble methods [13], are the most fast-growing areas in PHM. These methods map time series data from sensors to a hidden machine degradation process so as to estimate the RUL of the system at any given moment. In the era of big data and cloud computing, this ML-assisted RUL prediction approach is gaining popularity in both academic research and industrial practice. For example, Berghout et al. [14] proposed a new data-driven method based on the online sequential extreme learning machine algorithm for RUL prediction. Yan et al. [15] used the SVM classifier to evaluate the degradation stage of the bearings and achieved the best RUL results for different degradation stages through a hybrid degradation tracking model. Pan et al. [16] developed a performance degradation evaluation method based on the deep belief neural network and the self-organizing maps to denoise and merge multi-sensor vibration signals, then used the improved particle filtering optimization algorithm to predict the RUL. Yao et al. [17] combined an improved one-dimensional convolutional neural network (CNN) and a simple cyclic unit to overcome some shortcomings of traditional RUL prediction methods for rolling bearings. Liu et al. [18] proposed a double attention-based data-driven framework for RUL prediction, where a channel attention-based CNN and a transformer were applied to significant features. Overall, ML technologies, particularly deep neural networks that compose of multiple layers of nonlinear processing units [19] for recognizing complicated data patterns, have been successfully applied on RUL prediction in industrial PHM activities.

The degradation processes are non-Markov, that is, the current moment state depends on the previous moment. Because the mathematical basis of the recurrent neural network (RNN) can be regarded as Markov chain, in which the subsequent value is determined by the former in a certain probability. Thus, the RNN is regarded as a natural tool for time-series prediction in many studies. However, the gradient disappearance

* Corresponding author.
E-mail address: xxp15@tsinghua.org.cn (X. Xi).

https://doi.org/10.1016/j.ress.2023.109332
Received 16 September 2022; Received in revised form 9 March 2023; Accepted 29 April 2023
Available online 5 May 2023

will appear in traditional RNNs if the sequence is too long, i.e., the parameters can only capture the local relationship and cannot learn the long-term relationship. Long short-term memory (LSTM) [20] can avoid this problem to some extent due to the internal gating mechanism. Therefore, among a variety of neural networks, the LSTM learner has achieved very promising results by the virtue of its representation learning ability to time series. Yuan et al. [21] compared several variants of recurrent neural network, including traditional RNN, LSTM, gated recurrent unit LSTM, and AdaBoost-LSTM, and showed that LSTM was more effective than others for RUL prediction. Similarly, Wu et al. [22] demonstrated that LSTM is a natural fit for RUL prediction and could outperform other RNNs. Zhang et al. [23] proposed an LSTM-fusion architecture, where the LSTM was allowed to capture both local and global characteristics of the data from multiple sensors. These existing studies fully demonstrate that LSTM can play an important role in RUL prediction.

Nevertheless, a regular LSTM learner is prone to overfitting because of its long-term memory effect. Therefore, some researchers focused on combining LSTMs with other networks or analytical models or on improving the LSTM itself. Zheng et al. [24] proposed to use two LSTM layers, two feed-forward neural layers, and an output layer. Da Costa et al. combined an LSTM learner with a domain adversarial neural network (or the global attention layers) to learn the RUL relationships to time-series sensor data [25,26]. Wu et al. [27] presented a degradation-aware LSTM autoencoder scheme that could model degradation factors and explore latent variables, then predicted the RUL in multiple states. In addition, the LSTM RNN has been combined with elastic nets, empirical mode decomposition, clustering analysis, or CNNs to improve the accuracy and the robustness of RUL prediction [28–31]. Forward LSTM and backward LSTM were synthesized to generate a bidirectional LSTM, then integrated handshake rules, attention mechanisms, as well as change-point detection methods, had been proposed for the RUL prediction on physical systems [32–34]. These approaches utilized the additional information of equipment operation conditions, attention mechanisms and other neural network architectures to improve the prediction performance of LSTM learners.

However, the difference in time series length arising from various service times of equipment has not been considered adequately in the current LSTM-based RUL prediction studies. Either filling or truncating data will break the original data. Besides, the large difference in time series length will influence the adjustment of network weight, as the long-term memory of various sequence length may introduce a partially weakly dependent prediction series, and then affect the overall prediction accuracy of the learner. Therefore, it is necessary to concentrate on the short-term memory and to set the length of 'memory' dynamically. The short-term memory is usually offered by a neural network with time-delay structure, such as the nonlinear autoregressive exogenous (NARX) neural network [35] or the time delay neural network (TDNN) [36]. The inputs of these networks consist of finite time series, thus with less irrelevant information in long-term memory and producing more accurate predictions [37–39]. Rai et al. [40] proposed a data-driven prognostic approach based on the NARX neural network for RUL prediction of bearings, in which wavelet and MD-CUMSUM filter were used to process the raw signals and fuse features, respectively. Furthermore, the NARX neural network is also applied on stock price prediction [41], air pollutant prediction [42], and lithium-ion battery life prediction [43]. Compared with NARX, TDNN is more robust because its outputs at present moment are not utilized as inputs for subsequent moments. Zhu et al. [44] compared a TDNN with a multi-layer perceptron neural network for modeling of a wastewater system and found the TDNN is superior to the multi-layer perceptron network. Lipu et al. [45] used a TDNN to estimate the state of charge of lithium battery. Zhou et al. [46] presented a prediction approach based on physical aging model, autoregressive and moving average model, and a TDNN, and demonstrated that TDNN provided a higher level of predictive accuracy and robustness.

In addition to the aforementioned sequence's length of data and memory problem, the degradation characteristics of equipment usually are masked by a large amount of noisy sensor data, which makes it difficult to obtain a clear representation of equipment health state. The traditional one-stage methods establish the mapping between the original data and the RUL directly. However, due to the complex and changeable mapping relationship, the one-stage methods are difficult to obtain the ideal effect. From this perspective, the core of characterizing an equipment health degradation is to construct a health indicator (HI), which should be a smooth function of time and can represent the degradation trend intuitively and effectively. In this way, the complex mapping is broken down into two stages, from the raw data to HIs, then to the RULs. Most existing studies constructed a single HI based on feature fusion methods [40,47–49]. However, mechanical parts under various working conditions generally exhibit varying degradation behaviors due to the influence of external factors and the complexity of the parts themselves. To deal with these problems, a two-stage RUL prediction approach via LSTM and TDNN is proposed in this paper. The time-domain features extracted from sensor data are used as the inputs of an LSTM learner. Then, with the consideration of individual equipment degrading under different operation conditions and environments, a set of nonlinear HI functions are constructed to guide the LSTM learning. In order to refine and retain limited past information in the prediction process, a TDNN is utilized to map between HIs and RULs to mine the potential and time-varying state dependence relationship. Finally, the proposed LSTM-TDNN-based RUL prediction approach is tested on an accelerated degradation dataset of rolling element bearings.

Our LSTM approach to RUL prediction is novel because (1) a set of nonlinear HI functions are constructed to guide LSTM learners so as to accommodate the diverse degradation processes of industrial equipment; (2) an ensemble of LSTM learners are established from training, which compress or stretch time series to achieve the time scale alignment and maximize the retention of the original information; and (3) a time delay mechanism is introduced to fuse past feature information dynamically through a time window and to reduce the detrimental long-term memory effect on RUL prediction. Given multiple possible operating conditions of industrial equipment, these modeling strategies can provide more accurate RUL predictions.

The rest of this paper is organized as follows. Section 2 briefly introduces the elements of LSTM and TDNN. The proposed LSTM-TDNN-based RUL prediction approach is presented in detail in Section 3. Section 4 analyzes a public dataset of rolling element bearings to demonstrate the effectiveness of the proposed predictor and the superiority of its performance. The conclusions are drawn in Section 5.

2. Preliminaries

In this section, the elements of LSTM and TDNN are described briefly. For more information about these two neural network architectures and how they work, the readers are referred to [20,36]. An LSTM learner has several gating mechanisms with corresponding data compositions in each unit to handle the memory effect of a time series, while a TDNN learner has a relatively simple architecture for processing time series.

2.1. LSTM modeling

LSTM is a variant of RNNs that is capable of processing long-term information, and it is proposed by Hochreiter and Schmidhuber in 1997 [20]. Compared with traditional RNNs, the information of a time series at current and past moments can be selectively remembered, forgotten or updated by the gating mechanism employed in an LTSM unit, which consists of an input gate, an output gate, a forget gate and a memory cell. As a result, the problems of gradient disappearance and gradient explosion in a traditional RNN are avoided to a certain extent.

The mathematical formulas of these gates and memory cells are given below:

$$\begin{cases} c_t = f_t \otimes c_{t-1} + i_t \otimes \widetilde{c}_t \\ f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \\ i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \\ o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \\ \widetilde{c}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \\ h_t = o_t \otimes \tanh\left(c_t\right), \end{cases} \tag{1}$$

where c_t is the state of the memory cell at time t and $\widetilde{c}_t$ is the updated value of the state. Here, f_t, i_t and o_t are forget gate, input gate and output gate, respectively, and coefficients W_f, W_i, W_o and W_c, and b_f, b_i, b_o and b_c are the weights and biases used in constructing these gates. The function, $\sigma(\cdot)$, is the activation function, which generally employs a sigmoid function. Note that x_t is the input to an LSTM unit at time t, while h_t is the output of LSTM unit. As shown in Fig. 1, the forget gate f_t regulates the memory cell c_{t-1} to determine the extent to which a memory cell state can be retained from previous moments. The input gate i_t controls which information enters at time t and updates the cell state. The output gate c_t combines with o_t to produce the hidden state h_t. In our application, x_t is the raw time series data from sensors, h_t is the hidden memory state of the LSTM unit. A sequence of HIs are obtained through a series of weighted operations on h_t in a fully connected layer.

2.2. TDNN modeling

TDNN is a dynamic form of RNNs proposed by Waibel et al. in 1989 [36] and it has been successfully applied for modeling nonlinear systems. Both of its input and output are time series. Let HI_{t-1} be the input of the system at previous moment and d be the memory period that the network can retain historical data, then the mathematical formulation of TDNN can be expressed as

$$L_t = f\left[HI_{t-d}, \ldots, HI_{t-2}, HI_{t-1}\right], \tag{2}$$

where L_t is the output of the prediction system at a discrete time stamp t, and $f(\cdot)$ is a nonlinear mapping function represented by a neural network. It should be seen that L_t represents the one-step ahead prediction of system state using the system inputs within a time window. The details of nonlinear mapping of TDNN can be written as

$$L_t = f_o\left[b_o + \sum_{h=1}^{Nh} W_{ho} \cdot f_h\left(b_h + \sum_{a=1}^{d} W_{ih} \cdot HI_{t-a}\right)\right], \tag{3}$$

where $f_h(\cdot)$ and $f_o(\cdot)$ are the activation functions of the hidden and output layer, W_{ih} and W_{ho} are the network weight vectors, and b_h and b_o are the corresponding bias, respectively.

3. Proposed approach

This section describes the proposed LSTM-TDNN architecture for RUL predictor. The LSTM neural network is trained to obtain the HIs using the raw sensor data features from equipment condition monitoring. Then, the memories of past states of an equipment are stored in the LSTM cell for future state prediction. Now, the sequence of HIs are fed into the TDNN, and relying on the time delay effect of TDNN, RULs are predicted. Fig. 1 shows the architecture of the proposed LSTM-TDNN-based RUL predictor.

Table 1
Feature sets.

Time-domain features		
Max	Min	Peak
Peak to peak	Mean	Average amplitude
Root amplitude	Variance	Standard deviation
Rms	Kurtosis	Skewness
Shape factor	Peaking factor	Pulse factor
Margin factor	Kurtosis factor	Clearance factor

3.1. Stage one– From data to HI

3.1.1. Data preprocessing

To make full use of sensor data and to reduce the computational burden caused by noisy data, it is necessary to extract several key time-domain features from the raw data. These features are simple and general enough that they can be automatically extracted without any specially designed algorithm. In our application, 18 time-domain features of the monitoring time series are extracted at each sampling time stamp. These features are listed in Table 1.

Considering that these features may possess values over a wide range and this can lead to a problem of underfitting if they were directly used as the inputs of a neural network, we need to normalize each feature in the training sets to the interval [0,1] by the following equation:

$$x_{ij}^* = \frac{x_{ij} - x_{i,min}}{x_{i,max} - x_{i,min}}, \tag{4}$$

where x_{ij} is the ith feature of jth sample with $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. Here, m is the number of features and n is the number of samples. Let $x_{i,min}$ and $x_{i,max}$ be the minimum and maximum of value of the ith feature, respectively, and x_{ij}^* be the normalized value of x_{ij}. Furthermore, for each training time series, all of its normalized feature values are stored in the feature matrix $X^{(l)}$, and $l = 1, 2, \ldots, q$ denotes distinct training sets.

In addition, different training sets may have different time series lengths. When this time series is the series of an equipment degradation measures up to its failure, then the total series length corresponds to the life spans of the equipment. Using original RULs as the desired output of an LSTM neural network will result in a poor fitting. To deal with this issue, the RUL of each training time series is normalized by dividing it by its whole life span, i.e.,

$$\tilde{L}\left(t_k\right) = \frac{t_n - t_k}{t_n}, \tag{5}$$

where t_k is the sampling time at the kth time stamp with $1 \leqslant k \leqslant n$, and t_n is the life span of one training series. In other words, $\tilde{L}\left(t_k\right)$ can be interpreted as the percentage of remaining useful life at time t_k.

3.1.2. HI construction

Mechanical parts typically go through three health stages – normal, degradation, and failure – over their entire service period. The degradation rate is usually small at first, followed by an instantaneous jump or a fast increase. With this consideration, a group of nonlinear HI functions are proposed in this paper, which not only cover the impacts of varying working conditions and different equipment deterioration rates, but also have good representations in practical applications. These functions can be viewed as functions of latent states of equipment health in the proposed LSTM-TDNN-based RUL predictor.

The constructed HI is given by

$$HI_p^{(l)}(t) = V_p^{(l)}\left(\theta^{(l)}(t)\right), \tag{6}$$

where $p = 1, \ldots, u$ indicates individual HI functions, and $HI_p^{(l)}$ denotes the HI of the lth equipment, which is used in the training dataset, with the pth function. Function $V(\cdot)$ is the mapping function, and $\theta^{(l)}(t)$ is

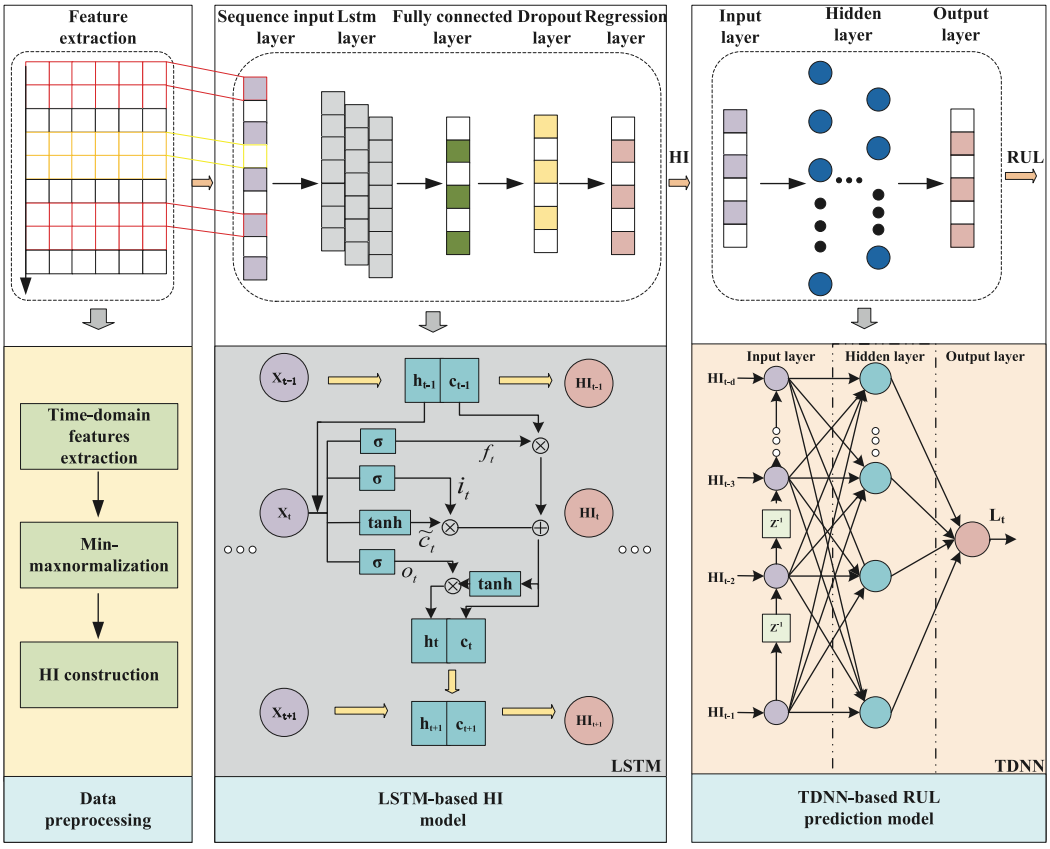

Fig. 1. Architecture of the proposed LSTM-TDNN-based RUL predictor.

the normalized sampling time. The normalized sampling time of the lth equipment at the kth moment i.e., $\theta^{(l)}(t_k)$ is given by:

$$\theta^{(l)}(t_k) = \frac{t_k}{t_n^{(l)}}, \tag{7}$$

where $t_n^{(l)}$ is the lifespan of the entire series of the lth training time series, and $\theta^{(l)}(t_k) \in (0, 1]$.

Four deterioration functions – the exponential function, power-law function, logarithmic function and composite function – are proposed. They are given by, respectively,

$$HI_p^{(l)}(t_k) = \begin{cases} 1 - e^{\theta^{(l)}(t_k)^r} & p = 1 \\ 1 - \theta^{(l)}(t_k)^r & p = 2 \\ 1 - \ln\left(1 + e^{\theta^{(l)}(t_k)^r} + \theta^{(l)}(t_k)^r\right) & p = 3 \\ 2 - \left(e^{\theta^{(l)}(t_k)^r} + \ln\left(1 + e^{\theta^{(l)}(t_k)^r} + \theta^{(l)}(t_k)^r\right)\right) & p = 4 \end{cases} \tag{8}$$

where $r \in N^*$. Parameter r is introduced to stretch the time scale and to make the shape of these functions more prominent. The value of this parameter should be set as small as possible while ensuring that the constructed HI represents the changes in degradation rate. In such way, the function $HI_p^{(l)}(t)$ over the whole time series lifespan is obtained. To standardize the HI, normalization is also employed,

$$\widetilde{HI}_p^{(l)}(t_k) = \frac{HI_p^{(l)}(t_k) - HI_{p,min}^{(l)}}{HI_{p,max}^{(l)} - HI_{p,min}^{(l)}}, \tag{9}$$

where $HI_{p,max}^{(l)}$ and $HI_{p,min}^{(l)}$ are the maximum and minimum of $HI_p^{(l)}(t)$, respectively.

Taking the feature set $X^{(l)}$ and the constructed HI set $\widetilde{HI}_p^{(l)}$ as the input and output of an LSTM learner respectively, the loss function, L_2, is minimized by adjusting the preset hyperparameters.

$$L_2\left(\widetilde{HI}_{po}^{(l)}, \widetilde{HI}_{pa}^{(l)}\right) = \sum_{k=1}^{n} \|\widetilde{HI}_{pa}^{(l)}(t_k) - \widetilde{HI}_{po}^{(l)}(t_k)\|_2, \tag{10}$$

where $\widetilde{HI}_{po}^{(l)}(t)$ is an output of the LSTM model, and $\widetilde{HI}_{pa}^{(l)}(t)$ is the constructed HI by Eqs. (7)–(9). To weaken the problem of different time series lifespans and to improve the transfer learning ability of trained networks, each training time series will establish an individual HI model

$$\widetilde{HI}_p^{(l)}(t) = F_p^{(l)}\left(X^{(l)}\right), \tag{11}$$

where $F_p^{(l)}$ is the nonlinear mapping function from features to the constructed pth HI of the lth training equipment.

3.2. Stage two– From HI to RUL

To map a normalized series of HIs to the RUL percentage, a mapping function G_p needs to be established. In our study, TDNN is such a mapping function, where the series of $\widetilde{HI}_p^{(l)}(t)$ of the training sets and the corresponding $\tilde{L}_p^{(l)}(t)$ are its inputs and outputs, respectively. To prevent the occurrence of over-fitting and improve accuracy, the whole training dataset is divided into a training subset, a validation subset and a testing subset. The termination condition of training process is set at the time when the error of the validation experiment is no longer reduced or even increased. After multiple training iterations, the LSTM-TDNN-based RUL predictor is obtained for several neural network architecture hyperparameter combinations, which include the time delay step d, the number of hidden layers M_2, the number of nodes in each layer Q_2, and the validation check P.

$$\tilde{L}_p^{(l)}(t) = G_p\left(\widetilde{HI}_p^{(l)}(t)\right). \tag{12}$$

$$\overline{HI}_p(t) = \frac{1}{q} \cdot \sum_{l=1}^{q} F_p^{(l)}\left(X^{(Test)}\right). \tag{13}$$

To summarize the proposed two-stage RUL prediction approach, the pseudo-code of the whole process is provided in Algorithm 1 and a flowchart is shown in Fig. 2. To use this predictor, the time-domain features of a testing time series are extracted. Feeding the feature set of the testing dataset, i.e., the normalized $X^{(Test)}$ into the LSTM-based HI model, the outputs of q LSTM learners are averaged to obtain the HIs, which are subsequently fed into the TDNN model to produce RUL predictions.

Algorithm 1 LSTM-TDNN-based RUL prediction approach

Training Phase

for each type of constructed HI **do**

STEP 1. Extracting and normalizing the time-domain features of the raw condition monitoring data of the training set, obtain $X^{(l)}$ by Eq. (4).

STEP 2. The preliminary HIs are constructed by Eqs. (7)-(9).

STEP 3. According to Eq. (11), taking $X^{(l)}$ and $\widetilde{HI}^{(l)}(t)$ as the input and output of the LSTM, and the optimal LSTM-based HI model is selected by fine-tuning the parameters and minimizing the loss function in Eq. (10).

end for

STEP 4. The output $\widetilde{HI}^{(l)}(t)$ and $\tilde{L}^{(l)}(t)$ of the training sets are seen as the inputs and outputs of the TDNN by Eq. (12), then the optimal TDNN-based RUL prediction model is obtained by training the network iteratively and minimizing the verified performance.

Testing Phase

for each type of HI **do**

STEP 1. Obtain $X^{(Test)}$ by preprocessing the raw condition monitoring data.

STEP 2. The optimal LSTM-based HI model is further modified by comparing the output HI of the validation set and the constructed HI by Eqs. (7)-(9).

STEP 3. Taking $X^{(Test)}$ as the input of the optimized LSTM-based HI model, then calculate the $\overline{HI}(t)$ by Eq. (13).

STEP 4. The $\tilde{L}_{pre}(t)$ is given by inputting $\overline{HI}(t)$ into the TDNN-based RUL prediction model.

end for

4. Case study

In this section, the XJTU-SY rolling element bearing dataset is employed to validate the effectiveness of the proposed two-stage LSTM-TDNN-based RUL predictor. Moreover, the superiority of the proposed approach is illustrated by comparing it with six other data-driven neural networks previously used for time series RUL prediction.

4.1. Data description and feature extraction

Fig. A.1 displays the block diagram of a rolling element bearing degradation testbed. The experimental platform is composed of an alternating current (AC) motor, a motor speed controller, a support shaft, two support bearings, a hydraulic loading system and tested bearings. Complete run-to-failure vibration data are collected by two single acceleration sensors of type PCB 352C33, which are fixed in horizontal and vertical directions of the tested bearings. The sampling frequency is 25.6 kHz, i.e., 32,768 data points are collected per minute. For a full description of the configuration of this testbed, please refer to [8,50].

There are three accelerated degradation conditions and five bearing systems are tested under each conditions. In our study, four bearing datasets are randomly selected as the training datasets and the remaining one is the testing dataset for each operating condition. The details are presented in Table 2. Fig. A.2 shows the vibration sensor data recorded over the whole life cycles of three different training datasets. As one can see from these figures, the trends of vibration magnitudes from horizontal and vertical signals of a single bearing are consistent; however, the degradation degrees at different times among these bearings are quite distinct from each other. To fully utilize these raw data, the feature data extracted from both vertical and horizontal vibration data are used as the inputs of the LSTM neural network, thus $m = 36$. As examples, the four horizontal time-domain features of bearing 1-1, including max, peak, peak to peak and rms before and after normalization are presented in Figs. 3 and 4, respectively. The amplitudes of these features gradually increase along the bearing running experiment, indicating that the bearing degradation process has been represented effectively. In addition, these features vary in the same range after normalization, making the LSTM allocate approximately equal weights for them to adjust parameters. Therefore, the utilization rate of LSTM-based HI model for various features is improved.

4.2. LSTM-TDNN construction

Both LSTM and TDNN are feed-forward neural networks, which consist of an input layer, at least one hidden layer and an output layer. The hidden layers of different networks are made up of multiple neurons of a certain kind. The neuron is the LSTM unit in the LSTM network, and is the ordinary artificial neuron in the TDNN. The value of the output layer is calculated by a series of weighted calculations of the output of the hidden layer. The number of hidden layers and the number of neurons in the network should be analyzed based on specific data. In general, the number of hidden layers and neurons has a positive correlation with the data volume.

For an LSTM learner, the labels of outputs are obtained from Eqs. (7)–(9), i.e., the training process is guided by a set of presumed system health degradation functions. The parameter r has been adjusted manually, but it needs to be kept low as much as possible. It is found that there is already a clear shape of HI curve for $r = 3$. As examples, the four constructed HI functions of bearing 1-1, 2-2, 3-3 are presented in Fig. 5. The slopes of these functions all change from small to big, but they still exhibit differences in curvature.

The number of hidden layers M_1, the number of nodes in each layer Q_1, initial learning rate L, learning rate decay D, max epoch B, and other hyperparameters of neural network are selected according to empirical knowledge. The Adam optimization algorithm [51] with a mini-batch size of 8 or 16 is applied to train the LSTM learner. It adapts the sparse gradient to alleviate the problem of gradient oscillation. A dropout layer is added after the LSTM layer with a dropout rate of S to ease the overfitting issue and to improve the generalization ability of the neural network. At the end, a fully connected layer is stacked to calculate softmax activation function by matrix multiplication so as to obtain HIs. Finally, the LSTM-based HI model is acquired by fine-tuning the above hyperparameters via minimizing the loss function of Eq. (10).

For the TDNN learner, the number of validation check P is set as 15 to prevent incomplete training and overtraining. The test results are obtained on the validation subset after the end of each epoch. As the epochs increase, if the error curve no longer decreases for 15 consecutive iterations, the training is stopped. Based on expert experience, P is usually set between 10 and 20. Here it is set to 15. Considering the learning ability of neural network and the total quantity of data, the time delay window d is set to be 5 and the number of hidden layers M_2 to be 2 or 3. Finally, the proportions of the sub-training, sub-validation and sub-testing sets are set to be 8:1:1. The Levenberg–Marquardt algorithm [52] is applied in TDNN training, which has the advantage of fast convergence to global optima

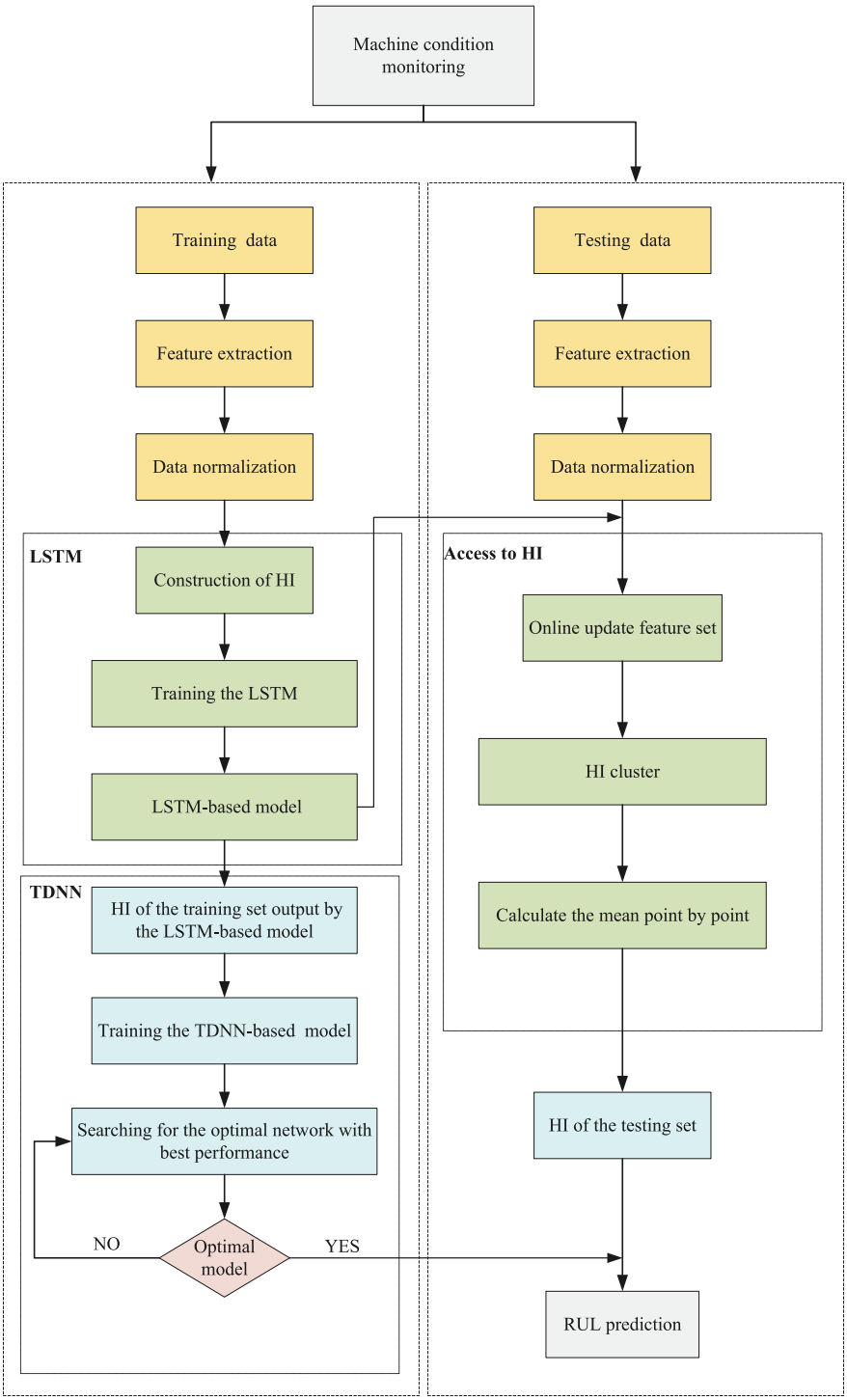

Fig. 2. Flowchart for the proposed LSTM-TDNN-based RUL predictor.

Table 2
Descriptions of XJTU-SY rolling element bearing datasets [8].

Operating condition	Rotating speed	Radial force/kN	Bearing dataset		
			Training dataset		Testing dataset
1	2100	12	Bearing 1-1 Bearing 1-4	Bearing 1-2 Bearing 1-5	Bearing 1-3
2	2250	11	Bearing 2-1 Bearing 2-3	Bearing 2-2 Bearing 2-4	Bearing 2-5
3	2400	10	Bearing 3-1 Bearing 3-4	Bearing 3-3 Bearing 3-5	Bearing 3-2

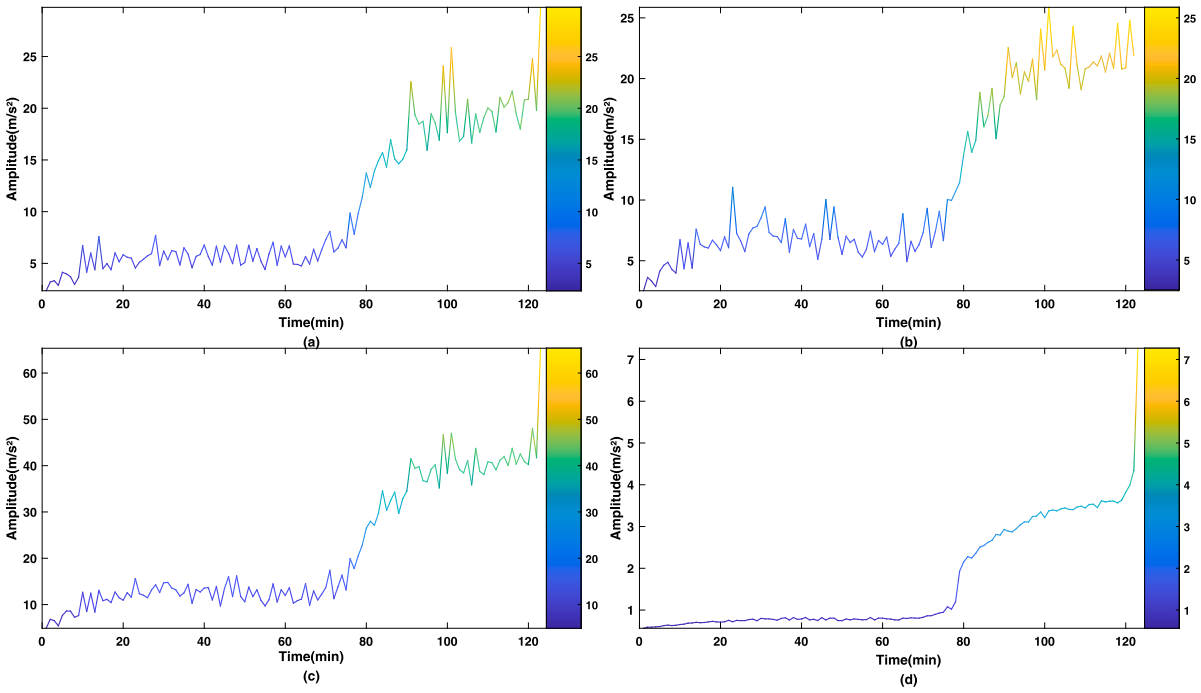

Fig. 3. The raw horizontal time-domain features of bearing 1-1 (a) max (b) peak (c) peak to peak (d) rms.

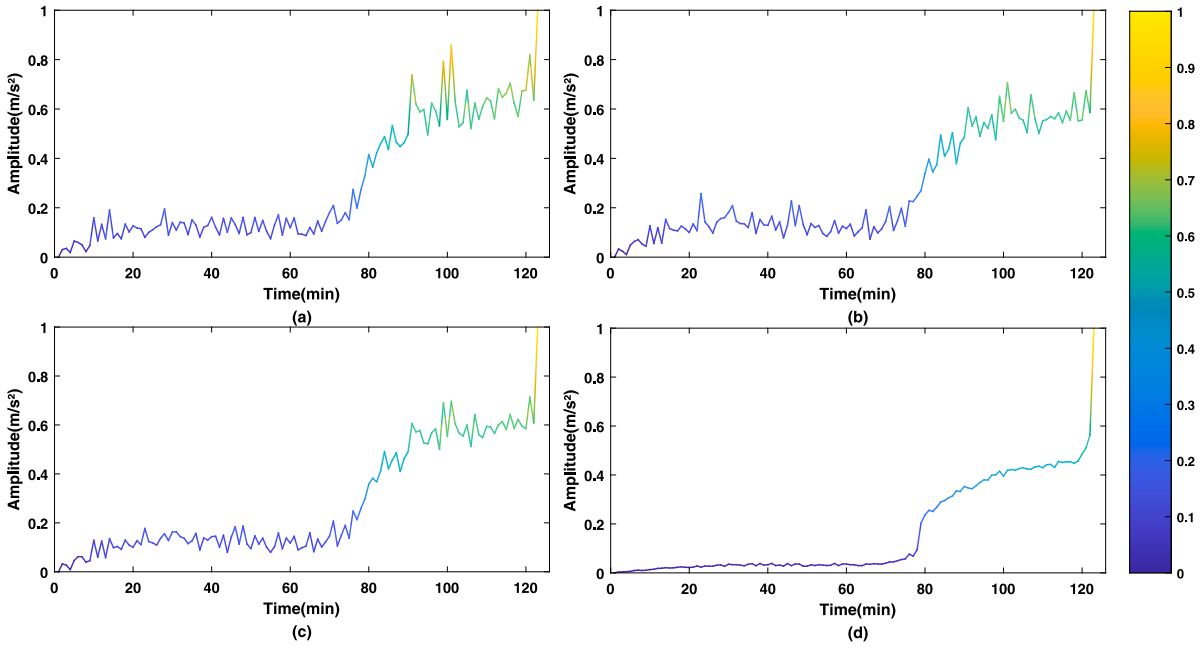

Fig. 4. The normalized horizontal time-domain features of bearing 1-1 (a) max (b) peak (c) peak to peak (d) rms.

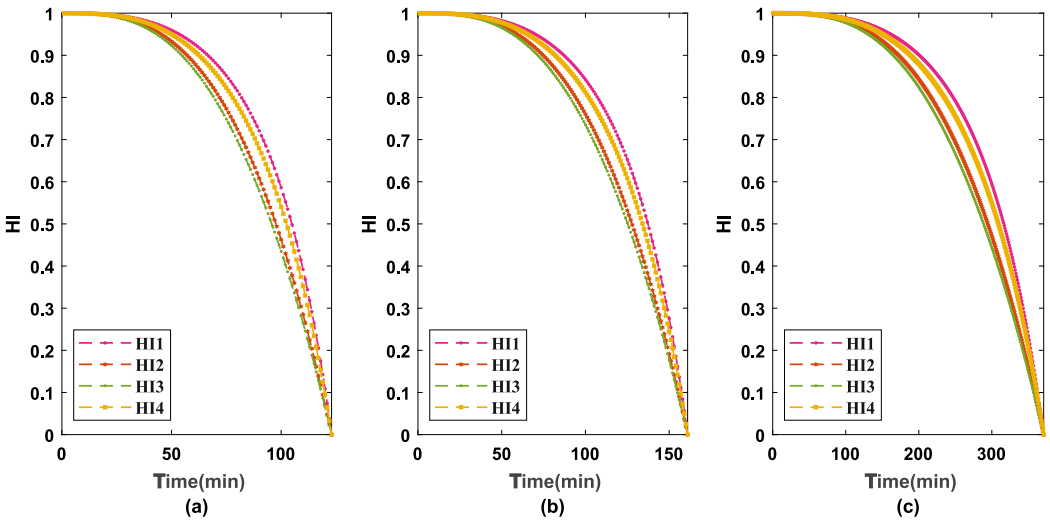

Fig. 5. Constructed HIs of 3 training bearings (a) bearing 1-1 (b) bearing 2-2 (c) bearing 3-3.

and achieving high precision solution. The configuration details of the LSTM-TDNN-based RUL predictors for the three bearing operation conditions are summarized in Tables 3–5. The size of each layer is shown in Tables A.1–A.3 in the Appendix.

The training processes of the LSTM and TDNN are completed by the computer, the trained network is considered reasonable when the error of training dataset decreases and converges with the increase of epochs. Generally, the establishment of the model can meet the above requirements by initializing learning rate L, max epoch B and other basic hyperparameters according to expert experience. Taking condition 1 as an example, the change of the loss function of the LSTM learner in the training process of bearing 1-2 with the number of epochs is shown in Table A.4 under HI_2. The loss function decreases and converges with the training. The errors in the training of the TDNN learner are shown in Fig. A.3. To further evaluate the performance of the RUL prediction model, the coefficient of determination, i.e., R is introduced, whose value is located in the [0,1]. The fitting ability of the model has a positive correlation with R. The value of the R of the training subset, validation subset, and testing subset are displayed in Fig. A.4, which are around 0.99. This reveals the good generalization ability of the RUL prediction model.

Table 3
Configuration of the LSTM-TDNN RUL predictor in condition 1.

Con. 1	LSTM						TDNN			
	M_1	Q_1	L	D	B	S	P	d	M_2	Q_2
HI_1	3	(4,3,3) (4,7,9) (4,5,5) (4,3,2)	0.1	0.2	600	0.001	15	5	2	(15,20)
HI_2	3	(4,3,3) (5,7,9) (8,6,3) (4,3,2)	0.1	0.2	600	0.0015	15	5	2	(15,25)
HI_3	3	(4,3,2) (5,7,7) (8,6,4) (4,5,5)	0.01	0.2	600	0.001	15	5	2	(15,20)
HI_4	3	(4,3,2) (5,7,7) (8,6,4) (4,5,5)	0.08	0.5	600	0.005	15	5	2	(15,18)

4.3. Evaluation metrics

The superiority of intelligent methods is difficult to prove in principle. The neural network is just a tool to deal with problems. The existing research is often based on its application scenario and application object, and relies on accuracy and other indicators to evaluate the effectiveness of the structure of the network. In this paper, the performance of LSTM-TDNN-based RUL predictor is evaluated by root mean square error (RMSE) and scoring function. RMSE is a commonly used evaluation metric in the field of PHM. With the RUL prediction results, the RMSE can be calculated by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{z=1}^{N}\left(\tilde{L}_{pre}(z) - \tilde{L}_{act}(z)\right)^2}, \tag{14}$$

where N is the number of the testing samples, $\tilde{L}_{pre}(z)$ and $\tilde{L}_{act}(z)$ are the predicted RUL and the actual RUL of the zth testing sample, respectively.

The scoring function used in this paper was originally given in the 2008 PHM Data Challenge [53], which is characterized by

$$\mathrm{score} = \begin{cases} \sum_{z=1}^{N} \exp\left(\dfrac{\tilde{L}_{act}(z) - \tilde{L}_{pre}(z)}{13}\right) - 1 & \tilde{L}_{pre}(z) \le \tilde{L}_{act}(z) \\ \sum_{z=1}^{N} \exp\left(\dfrac{\tilde{L}_{pre}(z) - \tilde{L}_{act}(z)}{10}\right) - 1 & \tilde{L}_{pre}(z) > \tilde{L}_{act}(z). \end{cases} \tag{15}$$

Inspired by [54], Fig. 6 shows RMSE and score as the functions of the error between $\tilde{L}_{pre}(z)$ and $\tilde{L}_{act}(z)$, in which the error ranges from −40 to 40. It is clearly seen that the bigger the absolute error value is, the higher RMSE and score are. Besides, because the score grows exponentially as the error increases, a large absolute error may result in a single outlier. However, when RMSE is applied, the effect from outliers is reduced because it is linear with absolute error. Nevertheless, both score and RMSE are small when the error is in interval [−1,1]. Thus, the influence of outliers on the RUL prediction is also reduced by the RUL normalization used in this paper. Generally speaking, $\tilde{L}_{pre}(z) \le \tilde{L}_{act}(z)$ is believed to achieve a conservative prediction in case of

Table 4
Configuration of the LSTM-TDNN RUL predictor in condition 2.

Con. 2	LSTM						TDNN			
	M_1	Q_1	L	D	B	S	P	d	M_2	Q_2
HI_1	3	(5,4,3) (4,4,3) (5,4,3) (4,3,2)	0.08	0.5	800	0.001	15	5	2	(15,20)
HI_2	3	(5,5,4) (5,4,3) (4,4,3) (3,3,2)	0.01	0.2	600	0.001	15	5	2	(15,25)
HI_3	3	(6,4,3) (4,4,3) (5,4,3) (4,3,2)	0.01	0.5	600	0.001	15	5	2	(15,20)
HI_4	3	(5,5,4) (4,4,3) (5,4,3) (4,3,2)	0.01	0.2	400	0.0015	15	5	2	(15,20)

Table 5
Configuration of the LSTM-TDNN RUL predictor in condition 3.

Con. 3	LSTM						TDNN			
	M_1	Q_1	L	D	B	S	P	d	M_2	Q_2
HI_1	3	(4,3,3) (5,3,2) (8,7,7) (3,4,4)	0.08	0.5	400	0.001	15	5	2	(23,25)
HI_2	3	(4,2,2) (2,3,5) (7,4,3) (3,3,4)	0.2	0.5	800	0.001	15	5	2	(28,25)
HI_3	3	(4,3,2) (2,3,5) (3,2,2) (3,4,5)	0.01	0.5	450	0.001	15	5	2	(20,20)
HI_4	3	(5,5,2) (2,3,5) (8,7,3) (3,4,4)	0.2	0.5	800	0.001	15	5	2	(15,12)

ensuring accuracy, because a maintenance plan can be implemented in advance to prevent the occurrence of catastrophic failures. According to Fig. 6, the score value on the left side of the origin is smaller than that on the right side under the same absolute error. Therefore, the smaller the score, the more conservative the prediction would be.

4.4. Experimental results

The comparison of the $\tilde{L}_{act}$ and $\tilde{L}_{pre}$ for the three testing bearings in the late stage are illustrated in Figs. 7–12. The predicted RULs deviate significantly from actual RULs at the beginning, but along with the bearing running experiment, as more useful information would be obtained, the predicted RUL will quickly get close to the actual RUL. In fact, the degradation of a bearing at its late life stage is significantly faster than its early life stage. It is essential to achieve an accurate estimation of the RUL percentage in the late stage to prevent a sudden catastrophic failure [40]. Our experimental results satisfy this requirement.

Prediction performance of the proposed predictor is provided in Table 6. The RMSE and score of bearing 3-2 are bigger than those of bearing 1-3 and 2-5. In particular, the predicted RUL of bearing 3-2 deviates greatly from the actual RUL twice as shown in Fig. 12,

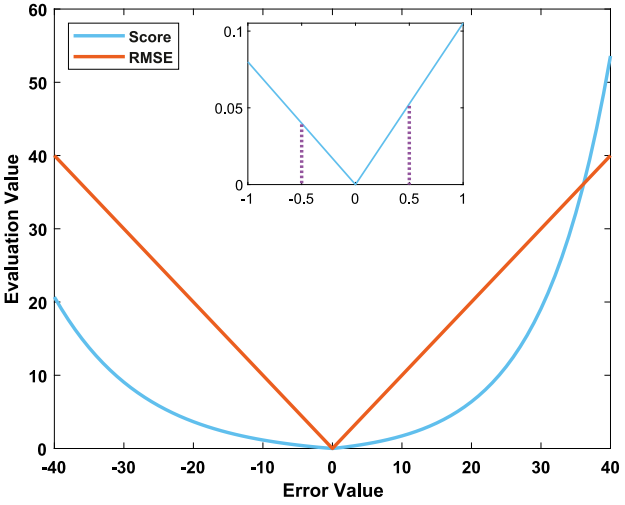

Fig. 6. Comparison between Score and RMSE under different error values.

indicating the existence of abnormal signals or faults in the raw vibration monitoring data. This is a general drawback of any data-driven methods, as they rely on the accuracy of monitoring data for accurate decision-making.

To further demonstrate the superiority of the proposed LSTM-TDNN predictor, six data-driven predictive approaches to time series prediction – LSTM, TDNN, LSTM-LSTM, TDNN-TDNN, TDNN-LSTM, and LSTM-NARX – are used for comparison. These networks can mine potential degradation information relying on memory structure, and are often used in the field of RUL prediction. However, there are differences in the length of memory and data processing. Moreover, the hyperparameters of these six prognostic networks, such as the number of epochs, the number of hidden layers, the number of time delay steps, and the number of validation checks, are all fine-turned and optimally selected after multiple iterations. Because the above models are constructed based on the same training data and empirical knowledge, the uncertainty is the same.

The comparisons of the prediction results and average training times of different neural network models are presented in Tables 6 and 7. The service time of the three testing bearing datasets varies with the operating conditions and the bearings themselves. Compared with the one-stage LSTM learner or one-stage TDNN learner, which building mapping relations from time domain features to RUL directly, two-stage approaches achieve better prediction results, although some of them require longer training times. This validates the importance of the constructed HI functions in guiding the process of RUL prediction. On one hand, it reduces the dimension of the data reasonably and maps the degradation features of different bearings into a unified space; on the other hand, it explores the hidden state of the degradation process effectively and makes the degradation trend of bearings more explicit.

Furthermore, among the five two-stage approaches, the proposed LSTM-TDNN-based predictor achieves the smallest RMSE and score, although it is more time-consuming in training compared with the TDNN-LSTM, TDNN-TDNN and LSTM-NARX. Based on the LSTM-TDNN model, the RMSEs of RUL for the three testing bearings are mostly around 0.02, and the maximum is less than 0.05. The average scores of these three test cases are 0.0630, 0.1385, and 1.7116, respectively.

Specifically, at the stage of obtaining HI, the TDNN-based models consume less time, but the outputs fluctuate greatly, especially when there is a large amount of monitoring data, just as the bearing 3-2. This is because there are no cell states in TDNN to store long sequences of information, only multiple delay units. Due to this unique structure, the calculation of TDNN is speeded up compared with long-memory neural networks. Besides, its 'memory' length changes dynamically with

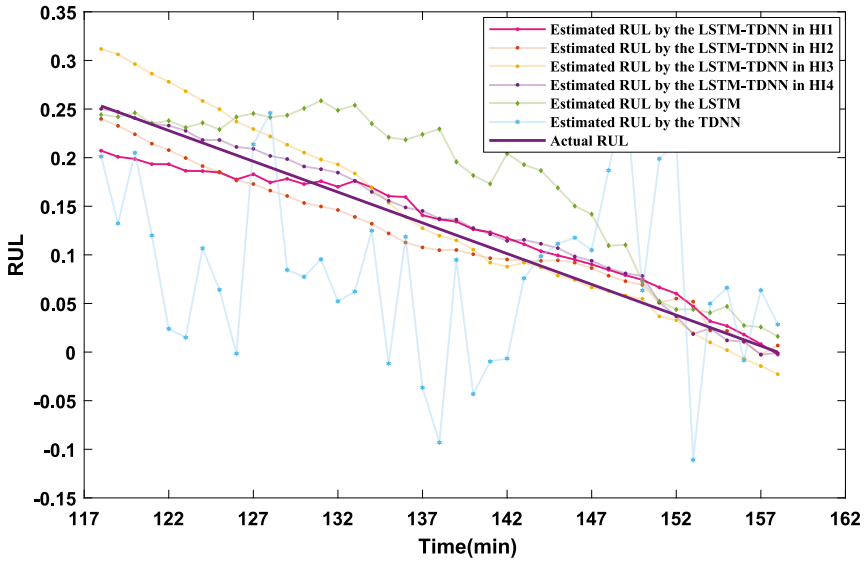

Fig. 7. Comparison of RUL prediction results for bearing 1-3 between one-stage and two-stage approaches.

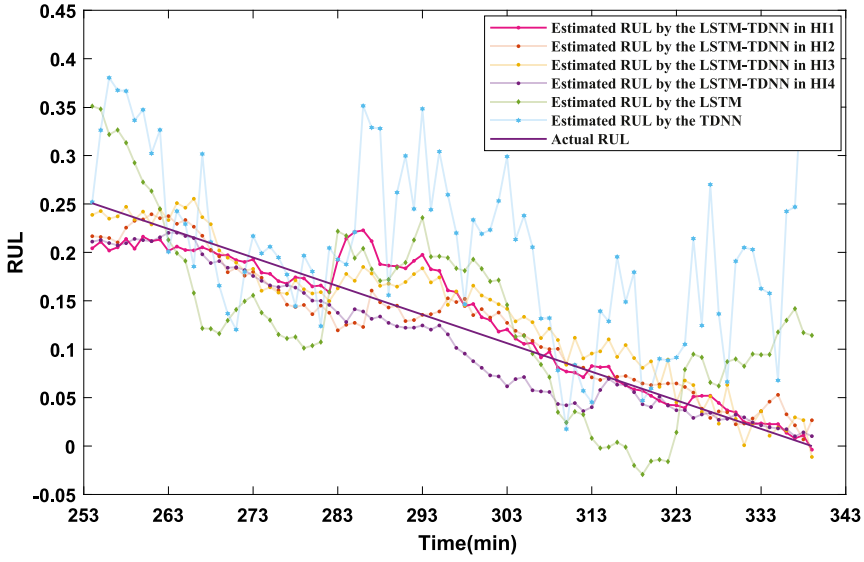

Fig. 8. Comparison of RUL prediction results for bearing 2-5 between one-stage and two-stage approaches.

Table 6
Prognostic performance comparisons of different predictors.

Testing bearing		Proposed		LSTM-LSTM		TDNN-LSTM		TDNN-TDNN		LSTM-NARX		LSTM		TDNN	
		RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
Bearing 1-3	HI_1	**0.0217**	**0.0674**	0.0291	0.0945	0.0904	0.3242	0.1234	0.3135	0.0675	0.1442	0.0602	0.2039	0.1147	0.3237
	HI_2	**0.0177**	**0.0545**	0.0214	0.0694	0.0263	0.0885	0.0344	0.0964	0.0460	0.1426				
	HI_3	**0.0277**	**0.0824**	0.0313	0.0899	0.0282	0.0851	0.1227	0.3779	0.2496	0.1333				
	HI_4	**0.0142**	**0.0479**	0.0172	0.0520	0.0262	0.0874	0.0854	0.2376	0.1083	0.2119				
Bearing 2-5	HI_1	**0.0245**	**0.1407**	0.0314	0.1830	0.0348	0.2201	0.1187	0.7324	0.0499	0.4008	0.0652	0.4623	0.1194	0.7789
	HI_2	**0.0195**	**0.1181**	0.0366	0.2177	0.0314	0.1833	0.0561	0.3813	0.0757	0.4335				
	HI_3	**0.0239**	**0.1698**	0.0364	0.1960	0.0327	0.2114	0.0718	0.4369	0.0313	0.2142				
	HI_4	**0.0225**	**0.1254**	0.0326	0.1898	0.0450	0.2556	0.0610	0.3610	0.0541	0.3583				
Bearing 3-2	HI_1	**0.0293**	**1.3031**	0.0299	1.3335	0.1200	6.1015	0.2294	10.8897	0.0600	2.9740	0.1135	4.6700	0.2531	11.5018
	HI_2	**0.0492**	**2.1436**	0.0515	2.2815	0.0834	3.8648	0.1262	5.9765	0.0504	2.3792				
	HI_3	**0.0313**	**1.4271**	0.0328	1.5885	0.1192	4.9011	0.2059	7.7929	0.0469	2.5969				
	HI_4	**0.0426**	**1.9727**	0.0440	2.0624	0.1271	5.2722	0.1721	2.5969	0.0854	3.3394				

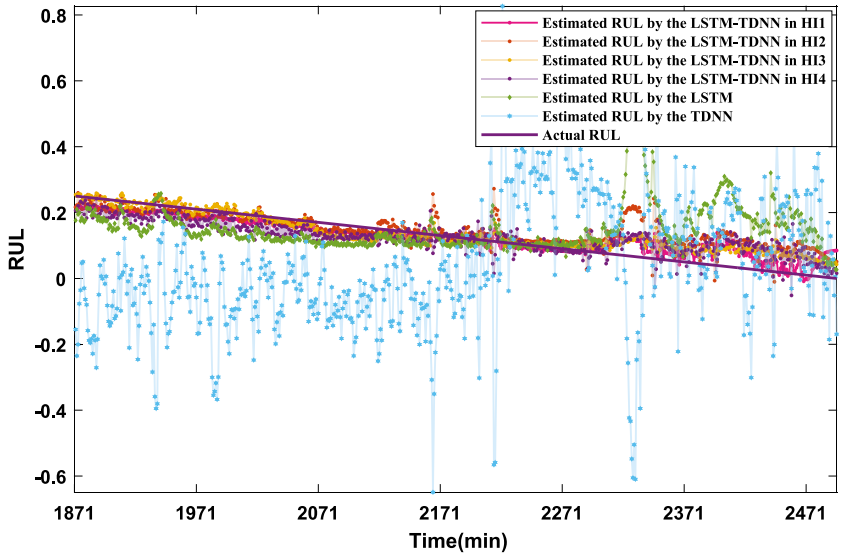

Fig. 9. Comparison of RUL prediction results for bearing 3-2 between one-stage and two-stage approaches.

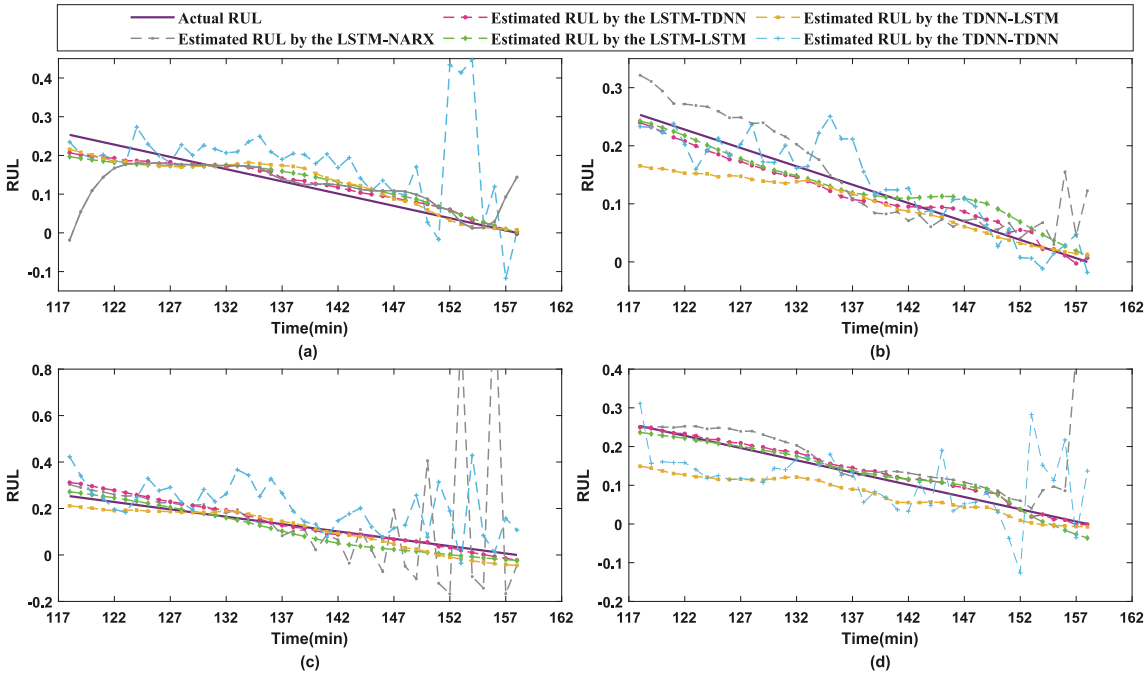

Fig. 10. RUL prediction results of bearing 1-3 by different predictors (a) RUL in HI_1 (b) RUL in HI_2 (c) RUL in HI_3 (d) RUL in HI_4.

Table 7
Comparison of average training time for seven different predictors.

	Operating condition	Proposed	LSTM-LSTM	TDNN-LSTM	TDNN-TDNN	LSTM-NARX	LSTM	TDNN
Training time (s)	Condition 1	60.25	94.25	51	7.38	60.51	33.39	16.11
	Condition 2	133.25	210.25	91	13.44	132.25	82.26	31.57
	Condition 3	487.26	807	431.56	58.72	477.03	281.46	131.29
	Total	680.76	1111.5	573.56	79.54	669.79	397.11	178.97

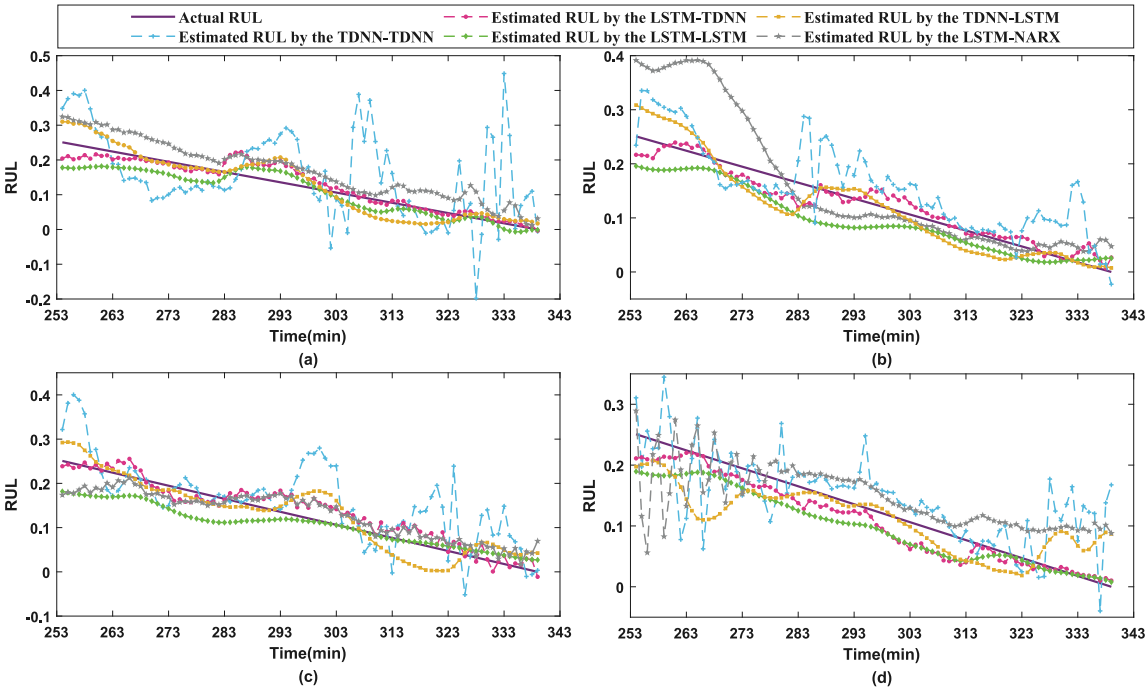

Fig. 11. RUL prediction results of bearing 2-5 by different predictors (a) RUL in HI_1 (b) RUL in HI_2 (c) RUL in HI_3 (d) RUL in HI_4.

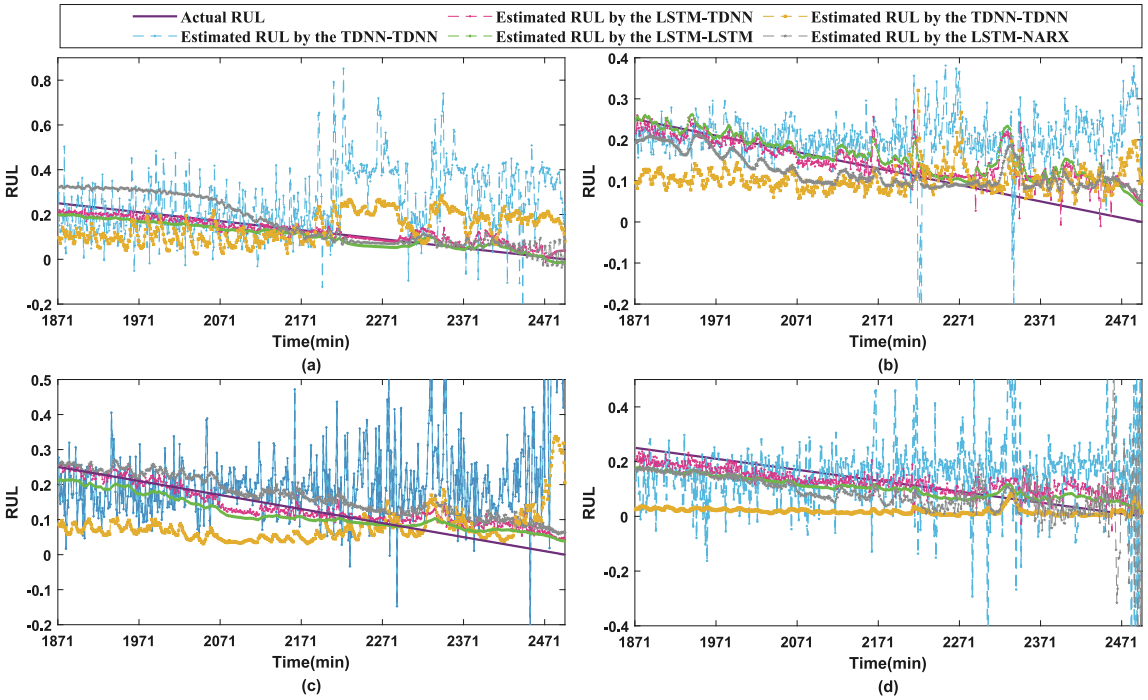

Fig. 12. RUL prediction results of bearing 3-2 by different predictors(a) RUL in HI_1 (b) RUL in HI_2 (c) RUL in HI_3 (d) RUL in HI_4.

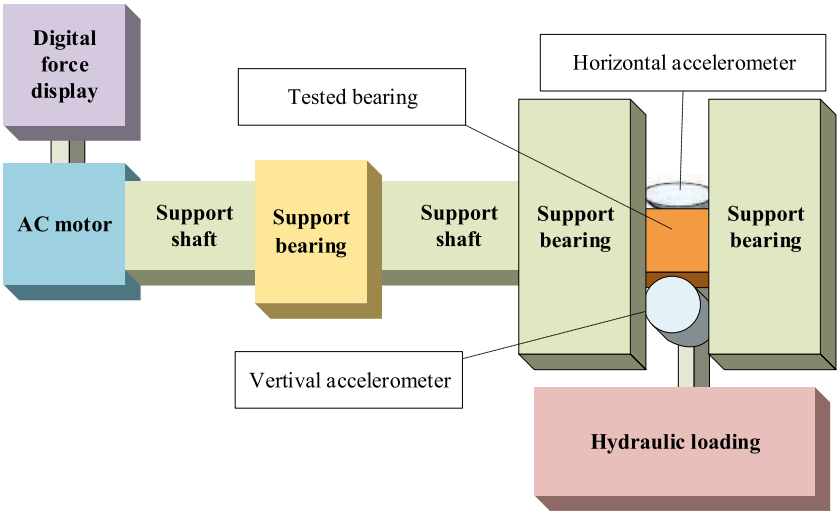

Fig. A.1. Test bed of the rolling element bearings [8,50].

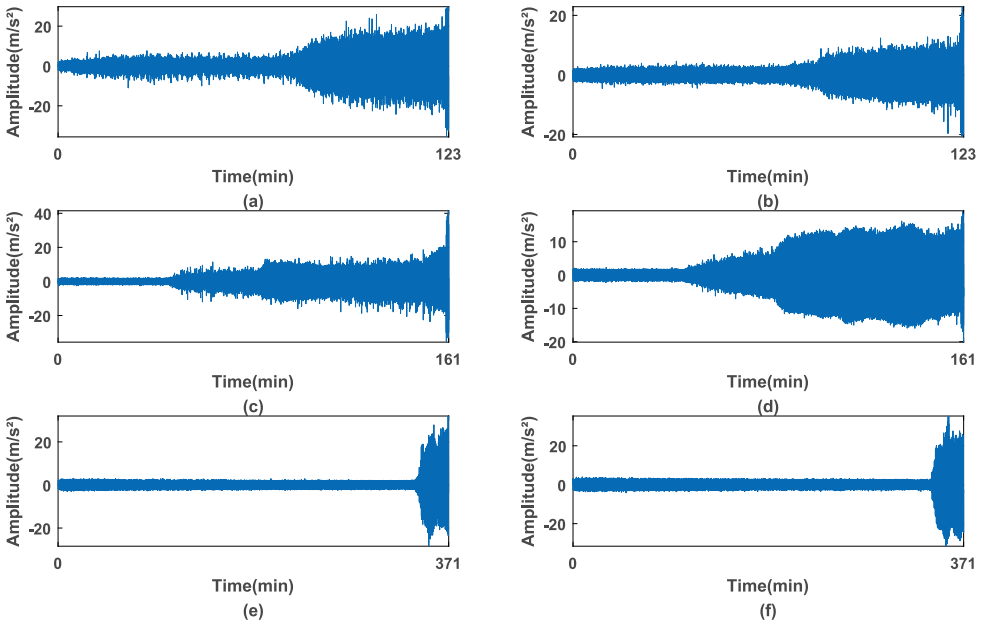

Fig. A.2. The raw vibration monitoring data of three training bearings (a) Horizontal vibration data of bearing 1-1 (b) Vertical vibration data of bearing 1-1 (c) Horizontal vibration data of bearing 2-2 (d) Vertical vibration data of bearing 2-2 (e) Horizontal vibration data of bearing 3-3 (f) Vertical vibration data of bearing 3-3.

the time delay step, i.e., the partial past information is fused only selectively within a time window. In LSTM neural networks, however, there are gating mechanism and state storage units, resulting in the capture of the long-term historical information. Thus, the HI outputs of the LSTM-based model have stronger correlation at different moments and there are no big jumps between different working conditions. The predicted RUL curves are also smooth, regardless of which learner is subsequently used to build the RUL prediction model, as shown in Figs. 10–12. As a result, the LSTM learner is superior to the TDNN learner in learning information from multi-dimensional features and obtaining smooth low-dimensional HI curve. As to the RUL prediction property of a learner, due to the long-term memory of LSTM, some irrelevant information that interferes with the prediction results may continue to be stored, resulting in a relatively large deviation between the predicted RUL and the actual RUL. However, adding a TDNN stage reduces the influence of redundant information, thus effectively increases the accuracy of prediction. In addition, compared with LSTM learner, TDNN learner achieves prediction one step ahead. Although NARX neural network can set dynamic delay step and achieve prediction ahead as well, it needs to rely on the output value fed back to the input, so the next output might be affected if there was an abnormal output value. In extreme, the prediction error could explode as the component or system approaches the failure moment.

A more detailed examination of Table 6 shows that the minimum RMSEs and scores for the three testing bearings are provided by the HIs constructed in composite, power-law and exponential function, respectively. Therefore, it is necessary to consider the difference in the degradation rate between bearings while predicting the RUL. More importantly, other approaches sometimes fail to achieve consistently accurate and conservative prediction results. For example, RMSE under HI_2 of bearing 3-2 based on the LSTM-LSTM model is bigger than that based on the LSTM-NARX model, but the score is smaller. Similarly,

Table A.1
The size of the LSTM-TDNN RUL predictor in condition 1.

Con. 1	LSTM					TDNN			
	Input layer	Hidden layer 1	Hidden layer 2	Hidden layer 3	Output layer	Input layer	Hidden layer 1	Hidden layer 2	Output layer
HI_1	1 × 36	36 × 4	4 × 3	3 × 3	3 × 1	1 × 1	1 × 15	15 × 20	20 × 1
		36 × 4	4 × 7	7 × 9	9 × 1				
		36 × 4	4 × 5	5 × 5	5 × 1				
		36 × 4	4 × 3	3 × 2	2 × 1				
HI_2	1 × 36	36 × 4	4 × 3	3 × 3	3 × 1	1 × 1	1 × 15	15 × 25	25 × 1
		36 × 5	5 × 7	7 × 9	9 × 1				
		36 × 8	8 × 6	6 × 3	3 × 1				
		36 × 4	4 × 3	3 × 2	2 × 1				
HI_3	1 × 36	36 × 4	4 × 3	3 × 2	2 × 1	1 × 1	1 × 15	15 × 25	25 × 1
		36 × 5	5 × 7	7 × 7	7 × 1				
		36 × 8	8 × 6	6 × 4	4 × 1				
		36 × 4	4 × 5	5 × 5	5 × 1				
HI_4	1 × 36	36 × 4	4 × 3	3 × 2	2 × 1	1 × 1	1 × 15	15 × 25	25 × 1
		36 × 5	5 × 7	7 × 7	7 × 1				
		36 × 8	8 × 6	6 × 4	4 × 1				
		36 × 4	4 × 5	5 × 5	5 × 1				

Table A.2
The size of the LSTM-TDNN RUL predictor in condition 2.

Con. 2	LSTM					TDNN			
	Input layer	Hidden layer 1	Hidden layer 2	Hidden layer 3	Output layer	Input layer	Hidden layer 1	Hidden layer 2	Output layer
HI_1	1 × 36	36 × 5	5 × 4	4 × 3	3 × 1	1 × 1	1 × 15	15 × 20	20 × 1
		36 × 4	4 × 4	4 × 3	3 × 1				
		36 × 5	5 × 4	4 × 3	3 × 1				
		36 × 4	4 × 3	3 × 2	2 × 1				
HI_2	1 × 36	36 × 5	5 × 5	5 × 4	4 × 1	1 × 1	1 × 15	15 × 25	25 × 1
		36 × 5	5 × 4	4 × 3	3 × 1				
		36 × 4	4 × 4	4 × 3	3 × 1				
		36 × 3	3 × 3	3 × 2	2 × 1				
HI_3	1 × 36	36 × 6	6 × 4	4 × 3	3 × 1	1 × 1	1 × 15	15 × 20	20 × 1
		36 × 4	4 × 4	4 × 3	3 × 1				
		36 × 5	5 × 4	4 × 3	3 × 1				
		36 × 4	4 × 3	3 × 2	2 × 1				
HI_4	1 × 36	36 × 5	5 × 5	5 × 4	4 × 1	1 × 1	1 × 15	15 × 20	20 × 1
		36 × 4	4 × 4	4 × 3	3 × 1				
		36 × 5	5 × 4	4 × 3	3 × 1				
		36 × 4	4 × 3	3 × 2	2 × 1				

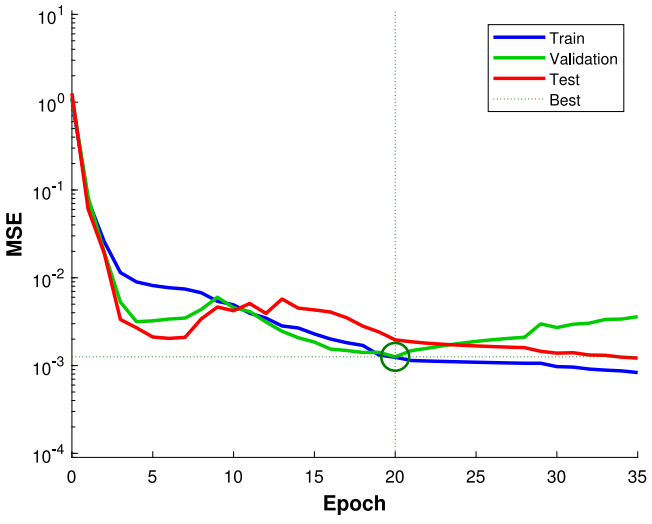

Fig. A.3. The training process of the TDNN learner in condition 1 under HI_2.

in terms of the LSTM-NARX model, the RMSE of bearing 1-3 under HI_1 is smaller than that under HI_3, but the score is bigger. With the TDNN-LSTM model, the RMSE under HI_4 of bearing 3-2 is bigger than that under HI_1, but the score is smaller. However, the proposed LSTM-TDNN predictor achieves the consistency at all three operating conditions.

In practice, an appropriate type of HI function at a certain working condition can be determined based on the monitoring data of existing failed equipment. After that, the RUL is further predicted conservatively based on the proposed RUL predictor, providing early warning for safe operation and for predictive maintenance.

5. Conclusions

In this paper, a novel two-stage LSTM-TDNN-based RUL predictor is proposed for complicated industrial equipment. A set of nonlinear HI functions are constructed to guide LSTM model building. Compared with the traditional feature fusion-based HI construction methods, the proposed approach considers various degradation rates and can be applied to a variety of working conditions. A series of LSTMs are used to build the mapping relations from individual time-series feature sets to

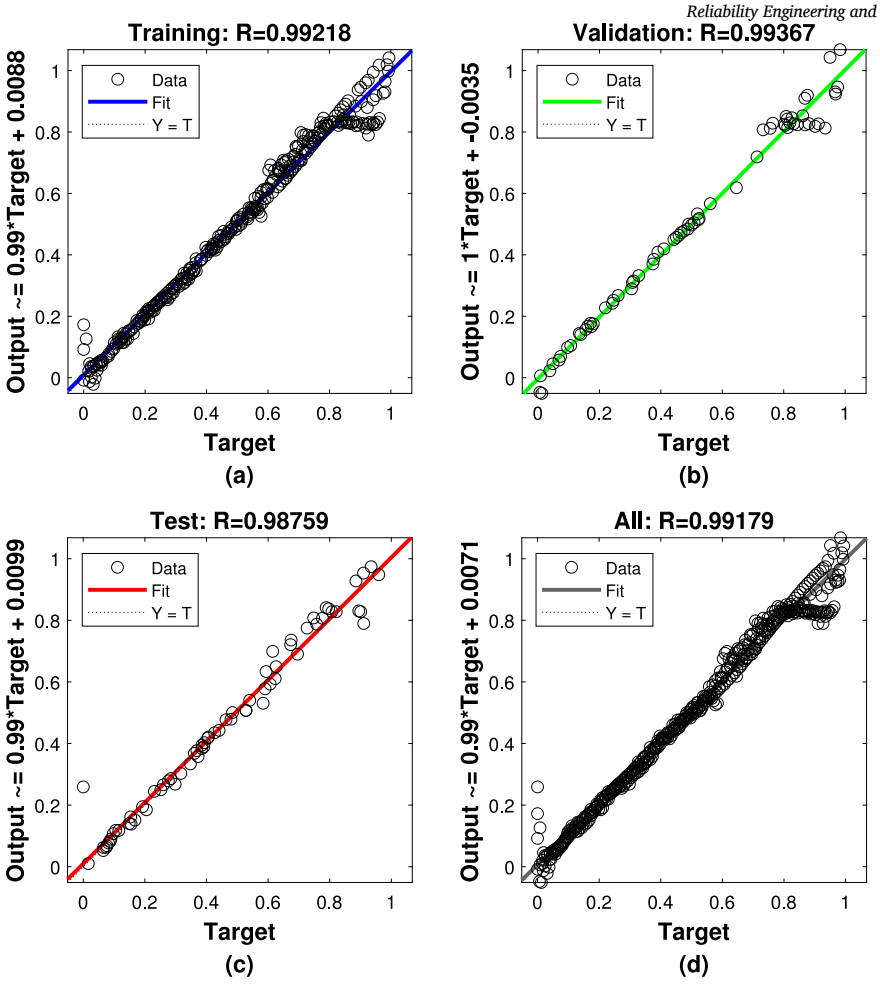

Fig. A.4. The coefficient of determination of the TDNN learner in condition 1 under HI_2.

Table A.3
The size of the LSTM-TDNN RUL predictor in condition 3.

Con. 3	LSTM					TDNN			
	Input layer	Hidden layer 1	Hidden layer 2	Hidden layer 3	Output layer	Input layer	Hidden layer 1	Hidden layer 2	Output layer
HI_1	1 × 36	36 × 4	4 × 3	3 × 3	3 × 1	1 × 1	1 × 23	23 × 25	25 × 1
		36 × 5	5 × 3	3 × 2	2 × 1				
		36 × 8	8 × 7	7 × 7	7 × 1				
		36 × 3	3 × 4	4 × 4	4 × 1				
HI_2	1 × 36	36 × 4	4 × 2	2 × 2	2 × 1	1 × 1	1 × 28	28 × 25	25 × 1
		36 × 2	2 × 3	3 × 5	5 × 1				
		36 × 7	7 × 4	4 × 3	3 × 1				
		36 × 3	3 × 3	3 × 4	4 × 1				
HI_3	1 × 36	36 × 4	4 × 3	3 × 2	2 × 1	1 × 1	1 × 20	20 × 20	20 × 1
		36 × 2	2 × 3	3 × 5	5 × 1				
		36 × 3	3 × 2	2 × 2	2 × 1				
		36 × 3	3 × 4	4 × 5	5 × 1				
HI_4	1 × 36	36 × 5	5 × 5	5 × 2	2 × 1	1 × 1	1 × 15	15 × 12	12 × 1
		36 × 2	2 × 3	3 × 5	5 × 1				
		36 × 8	8 × 7	7 × 3	3 × 1				
		36 × 3	3 × 4	4 × 4	4 × 1				

the HIs, aligning time scales of data and solving the problem of unequal length sequences to some extent. By introducing the TDNN stage, the historical information of HI in a finite time window is fused to achieve further refinement of prediction. The proposed predictor is compared with six other data-driven ML approaches on the XJTU-SY rolling element bearing accelerated life cycle datasets. Our experimental results reveal that the proposed predictor has a clear advantage in accuracy and conservativeness in RUL prediction. This is invaluable to proper maintenance scheduling and equipment service life extension.

Although satisfactory results have been obtained in this paper, the training of LSTM-TDNN model will be affected by the service time of the application object. For example, the HI of the testing individual

Table A.4
RMSE and Loss of the minibatch of the LSTM learner during training.

	Epoch												
	1	50	100	150	200	250	300	350	400	450	500	550	600
RMSE	0.96	0.07	0.04	0.04	0.08	0.03	0.01	0.01	0.01	0.08	8.54e−3	8.05e−3	8.00e−3
Loss	0.5	2.6e−3	7.5e−4	8.3e−4	3.3e−3	4.2e−4	7.3e−5	6.1e−5	5.2e−5	3.1e−3	3.6e−5	3.2e−5	3.1e−3

output by the LSTM-based model deviates from the HI constructed by the functions when there is large difference in actual life cycle between the testing individual and the training individual, thus affecting the prediction accuracy of the RUL. Therefore, it is still challenging under small sample to apply a common transfer learning technique to migrate the RUL predictor from one equipment to another one with different application scenarios and/or different machinery physics. Effective transferring learning approaches to equipment RUL prediction will be researched in the future.

CRediT authorship contribution statement

Huixin Zhang: Writing – original draft, Validation, Software, Investigation. **Xiaopeng Xi:** Writing – review & editing, Supervision, Funding acquisition. **Rong Pan:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

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

Data availability

Data will be made available on request

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62103244), Shandong Provincial Natural Science Foundation (ZR2021QF087), and the Qingdao Postdoctoral Applied Research Project. The third author also acknowledges the support from National Science Foundation, United States, grant 2134409.

Appendix

Fig. A.1 shows the test bad of the rolling element bearings. Fig. A.2 displays the raw vibration monitoring data of three training bearings. Tables A.1–A.3 show the size of the proposed LSTM-TDNN RUL predictor in conditions 1-3. Table A.4 is the RMSE and Loss of the minibatch of the LSTM learner during training. The errors in the training of the TDNN learner are shown in Fig. A.3. Fig. A.4 displays the coefficient of determination of the TDNN learner in condition 1 under HI_2.

References

[1] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mech Syst Signal Process 2018;104:799–834.
[2] Rigamonti M, Baraldi P, Zio E, Roychoudhury I, Goebel K, Poll S. Ensemble of optimized echo state networks for remaining useful life prediction. Neurocomputing 2018;281:121–38.
[3] Xi X, Zhou D, Chen M, Balakrishnan N, Zhang H. Remaining useful life prediction for multivariable stochastic degradation systems with non-Markovian diffusion processes. Qual Reliab Eng Int 2020;36(4):1402–21.
[4] Zhang Z, Si X, Hu C, Lei Y. Degradation data analysis and remaining useful life estimation: A review on Wiener-process-based methods. European J Oper Res 2018;271(3):775–96.
[5] Wang B, Lei Y, Yan T, Li N, Guo L. Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. Neurocomputing 2020;379:117–29.
[6] Zhu L, Liu C. Recent progress of chatter prediction, detection and suppression in milling. Mech Syst Signal Process 2020;143:106840.
[7] Pecht M. Prognostics and health management of electronics. In: Encyclopedia of structural health monitoring. 2020.
[8] Wang B, Lei Y, Li N, Li N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. IEEE Trans Reliab 2018;69(1):401–12.
[9] Noble WS. What is a support vector machine? Nature Biotechnol 2006;24(12):1565–7.
[10] Mitra A, Pan R. Early prediction of lithium-ion battery cycle life by machine learning methods. In: 2022 proceedings of annual reliability and maintainability symposium. IEEE; 2021.
[11] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[12] Pei H, Hu C, Si X, Zhang J, Pang Z, Zhang P. Review of machine learning based remaining useful life prediction methods for equipment. J Mech Eng 2019;55(8):1–13.
[13] Wang L, Lu D, Wang X, Pan R, Wang Z. Ensemble learning for predicting degradation under time-varying environment. Qual Reliab Eng Int 2020;36(4):1205–23.
[14] Berghout T, Mouss LH, Kadri O, Saïdi L, Benbouzid M. Aircraft engines remaining useful life prediction with an improved online sequential extreme learning machine. Appl Sci 2020;10(3):1062.
[15] Yan M, Wang X, Wang B, Chang M, Muhammad I. Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. ISA Trans 2020;98:471–82.
[16] Pan Y, Hong R, Chen J, Wu W. A hybrid DBN-SOM-PF-based prognostic approach of remaining useful life for wind turbine gearbox. Renew Energy 2020;152:138–54.
[17] Yao D, Li B, Liu H, Yang J, Jia L. Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit. Measurement 2021;175:109166.
[18] Liu L, Song X, Zhou Z. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. Reliab Eng Syst Saf 2022;221:108330.
[19] Si X, Wang W, Hu C, Zhou D. Remaining useful life estimation–a review on the statistical data driven approaches. European J Oper Res 2011;213(1):1–14.
[20] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[21] Yuan M, Wu Y, Lin L. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. In: 2016 IEEE international conference on aircraft utility systems. AUS, IEEE; 2016, p. 135–40.
[22] Wu Y, Yuan M, Dong S, Lin L, Liu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 2018;275:167–79.
[23] Zhang Y, Hutchinson P, Lieven NA, Nunez-Yanez J. Remaining useful life estimation using long short-term memory neural networks and deep fusion. IEEE Access 2020;8:19033–45.
[24] Zheng S, Ristovski K, Farahat A, Gupta C. Long short-term memory network for remaining useful life estimation. In: 2017 IEEE international conference on prognostics and health management. ICPHM, IEEE; 2017, p. 88–95.
[25] Da Costa PRDO, Akçay A, Zhang Y, Kaymak U. Remaining useful lifetime prediction via deep domain adaptation. Reliab Eng Syst Saf 2020;195:106682.
[26] Da Costa PRDO, Akçay A, Zhang Y, Kaymak U. Attention and long short-term memory network for remaining useful lifetime predictions of turbofan engine degradation. Int J Progn Health Manag 2019;10(4).
[27] Wu J, Wu M, Chen Z, Li X, Yan R. Degradation-aware remaining useful life prediction with LSTM autoencoder. IEEE Trans Instrum Meas 2021;70:1–10.
[28] Liu Z, Meng X, Wei H, Chen L, Lu B, Wang Z, et al. A regularized LSTM method for predicting remaining useful life of rolling bearings. Int J Autom Comput 2021;18(4):581–93.
[29] Guo R, Wang Y, Zhang H, Zhang G. Remaining useful life prediction for rolling bearings using EMD-RISI-LSTM. IEEE Trans Instrum Meas 2021;70:1–12.
[30] Liu J, Lei F, Pan C, Hu D, Zuo H. Prediction of remaining useful life of multi-stage aero-engine based on clustering and LSTM fusion. Reliab Eng Syst Saf 2021;214:107807.
[31] Dong Y, Xia T, Wang D, Fang X, Xi L. Infrared image stream based regressors for contactless machine prognostics. Mech Syst Signal Process 2021;154:107592.
[32] Elsheikh A, Yacout S, Ouali MS. Bidirectional handshaking LSTM for remaining useful life prediction. Neurocomputing 2019;323:148–56.

[33] Shah SRB, Chadha GS, Schwung A, Ding SX. A sequence-to-sequence approach for remaining useful lifetime estimation using attention-augmented bidirectional LSTM. Intell Syst Appl 2021;10:200049.
[34] Shi Z, Chehade A. A dual-LSTM framework combining change point detection and remaining useful life prediction. Reliab Eng Syst Saf 2021;205:107257.
[35] Lin T, Horne BG, Tino P, Giles CL. Learning long-term dependencies in NARX recurrent neural networks. IEEE Trans Neural Netw 1996;7(6):1329–38.
[36] Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ. Phoneme recognition using time-delay neural networks. IEEE Trans Acoust Speech Signal Process 1989;37(3):328–39.
[37] Zemouri R, Racoceanu D, Zerhouni N. Recurrent radial basis function network for time-series prediction. Eng Appl Artif Intell 2003;16(5–6):453–63.
[38] Soualhi A, Razik H, Clerc G. Data driven methods for the prediction of failures. In: 2019 IEEE 12th international symposium on diagnostics for electrical machines, power electronics and drives. SDEMPED, IEEE; 2019, p. 474–80.
[39] Yilboga H, Eker ÖF, Güçlü A, Camci F. Failure prediction on railway turnouts using time delay neural networks. In: 2010 IEEE international conference on computational intelligence for measurement systems and applications. IEEE; 2010, p. 134–7.
[40] Rai A, Upadhyay S. The use of MD-CUMSUM and NARX neural network for anticipating the remaining useful life of bearings. Measurement 2017;111:397–410.
[41] Dhafer AH, Mat Nor F, Alkawsi G, Al-Othmani AZ, Ridzwan Shah N, Alshanbari HM, et al. Empirical analysis for stock price prediction using NARX model with exogenous technical indicators. Comput Intell Neurosci 2022;2022.
[42] Adnane A, Leghrib R, Chaoufi J, Chirmata A. Prediction of PM10 concentrations in the city of Agadir (Morocco) using non-linear autoregressive artificial neural networks with exogenous inputs (NARX). Mater Today: Proc 2022;52:146–51.
[43] Khaleghi S, Hosen MS, Karimi D, Behi H, Beheshti SH, Van Mierlo J, et al. Developing an online data-driven approach for prognostics and health management of lithium-ion batteries. Appl Energy 2022;308:118348.
[44] Zhu J, Zurcher J, Rao M, Meng M. An on-line wastewater quality predication system based on a time-delay neural network. Eng Appl Artif Intell 1998;11(6):747–58.
[45] Lipu H, Hannan M, Hussain A, Ayob A, Saad MH, Muttaqi KM. State of charge estimation in lithium-ion batteries: A neural network optimization approach. Electronics 2020;9(9):1546.
[46] Zhou D, Al-Durra A, Zhang K, Ravey A, Gao F. Online remaining useful lifetime prediction of proton exchange membrane fuel cells using a novel robust methodology. J Power Sources 2018;399:314–28.
[47] Yang F, Habibullah MS, Shen Y. Remaining useful life prediction of induction motors using nonlinear degradation of health index. Mech Syst Signal Process 2021;148:107183.
[48] Zhao S, Zhang Y, Wang S, Zhou B, Cheng C. A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method. Measurement 2019;146:279–88.
[49] Wen P, Zhao S, Chen S, Li Y. A generalized remaining useful life prediction method for complex systems based on composite health indicator. Reliab Eng Syst Saf 2021;205:107241.
[50] Lei Y, Han T, Wang B, Li N, Yan T, Yang J. XJTU-SY rolling element bearing accelerated life test datasets: a tutorial. J Mech Eng 2019;55(16):1–6.
[51] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.
[52] Moré JJ. The Levenberg–Marquardt algorithm: implementation and theory. In: Numerical analysis. Springer; 1978, p. 105–16.
[53] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 IEEE international conference on prognostics and health management. ICPHM, IEEE; 2008, p. 1–9.
[54] Wang B, Lei Y, Li N, Yan T. Deep separable convolutional network for remaining useful life prediction of machinery. Mech Syst Signal Process 2019;134:106330.