Applied Soft Computing Journal 86 (2020) 105919

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

A novel deep learning method based on attention mechanism for bearing remaining useful life prediction

Yuanhang Chen, Gaoliang Peng *, Zhiyu Zhu, Sijue Li

State Key Laboratory of Robotics and System, Harbin Institute of Technology, No. 92 Xidazhi Street, Harbin 150001, Heilongjiang Province, China

ARTICLE INFO

Article history:
Received 8 April 2019
Received in revised form 23 September 2019
Accepted 3 November 2019
Available online 5 November 2019

Keywords:
Remaining useful life prediction
Recurrent neural network
Attention mechanism

ABSTRACT

Rolling bearing is a key component in rotation machine, whose remaining useful life (RUL) prediction is an essential issue of constructing condition-based maintenance (CBM) system. However, recent data-driven approaches for bearing RUL prediction still require prior knowledge to extract features, construct health indicate (HI) and set up threshold, which is inefficient in the big data era. In this paper, a pure data-driven method for bearing RUL prediction with little prior knowledge is proposed. This method includes three steps, i.e., features extraction, HI prediction and RUL calculation. In the first step, five band-pass energy values of frequency spectrum are extracted as features. Then, a recurrent neural network based on encoder–decoder framework with attention mechanism is proposed to predict HI values, which are designed closely related with the RUL values in this paper. Finally, the final RUL value can be obtained via linear regression. Experiments carried out on the dataset from PRONOSTIA and comparison with other novel approaches demonstrate that the proposed method achieves a better performance.

1. Introduction

As one of the most critical components, rolling element bearing is widely used to evaluate the statement of rotating machine. Any unexpected bearing failures would deflect the mechanical system from normal state, such as precision loss, productivity reduction and even the increase of safety risks [1–5]. To employ condition-based maintenance (CBM) strategy on rotating machine to avoid such disaster, bearing remaining useful life (RUL) estimation is one of the major tasks.

Generally, RUL prediction methods could be grouped into two main categories, i.e. model-based approaches and data-driven approaches [6]. The core of model-based approaches is to build a mathematical model to accurately describe the degradation of machinery. However, the constructing process requires not only the parameters of the real engineering system after a series of measurements, but also extensive prior knowledge about the systems. Actually, few machineries can be accurately simulated by simple mathematical models for the complexity of real world. Nowadays, model-based approaches for predicting a general trend of the degradation of machinery are still helpful, including particle filter [7] , Eyring model [8] , Weibull distribution [9], etc. On the other hand, data-driven methods attempt to learn the machinery degradation patterns based on historical collected data. Thanks to the strong learning ability, these methods are capable of revealing the underlying correlations and causalities between the phenomenon (the collected data) and the reason (the statement of the corresponding system), especially the complex ones that hardly can be described by manual mathematical model. Because of this characteristic, more and more data-driven approaches in the field of machinery prognostics are proposed and achieved better prognostic results. A comprehensive review of statistical data-driven approaches is presented by Si et al. [10]. Lei et al. [11] built a data-driven model based on artificial neural network to predict bearing RUL. In [12], Huang et al. suggested that a novel health indicator (HI) called minimum quantization error obtained by a self-organizing map (SOM), is then helpful to train back propagation neural networks for degradation prediction model. Chen et al. [13] developed a prognostic method using adaptive neuro-fuzzy inference systems and high order particle filtering. Loutas et al. [14] proposed a data-driven approach for bearing RUL estimation based on support vector regression (SVR), which utilizes multiple statistical features from time-domain, frequency domain and time-scale domain. And more data-driven approaches are introduced briefly in a recent review paper [15].

There is no doubt that, when faced with enormous amount of collected data, data-driven approaches based on deep learning (DL) show more effective processing capacity and achieve more excellent performance, especially in the field of computer vision, natural language processing, speech processing, etc. [16–18]. And

* Corresponding author.
E-mail addresses: cyh.wne@gmail.com (Y. Chen), pgl7782@hit.edu.cn (G. Peng), zbzhzhy@gmail.com (Z. Zhu), lisijue@hit.edu.cn (S. Li).

https://doi.org/10.1016/j.asoc.2019.105919
1568-4946/© 2019 Elsevier B.V. All rights reserved.

some deep learning techniques have already found their way into machine health monitoring systems. Zhu et al. proposed a stacked autoencoder (SAE) based DNN (Deep Neural Network) for hydraulic pump fault diagnosis that uses frequency features generated by Fourier transform [19]. Liu et al. uses normalized spectrum generated by short-time Fourier transform (STFT) of sound signal as inputs of a 2-layer SAE based DNN. Some researchers [20,21] feed multi-domain statistical features including time domain features, frequency domain features and time-frequency domain features into SAE as a way of feature fusion. Zhang et al. [22,23] proposed a novel convolution neural network (CNN) to make a successful bearing fault diagnosis directly on vibration signals. Furthermore, other difficult bearing fault diagnosis problems can be solved by more complex CNNs [24–26]. Although more and more DL-based approaches are employed to deal with fault diagnose problems in mechanical system, few successful cases can be found in addressing prognosis problem. Recurrent neural network (RNN), instead of CNN, is a smart choice for seeking underlying knowledge from historical data. Malhi et al. [27] proposed a competitive learning-based approach based on RNN for long-term prognostics of machine health statement, where vibration signals collected from a defected rolling bearing are pre-processed with the continuous wavelet transform and used as the model input. Besides, a long short-term memory (LSTM) based neural network scheme was proposed by Yuan et al. [28] for RUL estimation of aero-engines, in the cases of complicated operations, hybrid faults and strong noises. LSTM was also utilized by Zhao et al. [29] for a tool wear health monitoring task.

Though many works mentioned above have achieved good results, few of them are purely data-driven. In order to make an accurate prognosis, two main steps are still essential: health indicator calculation and bearing RUL prediction. For now, manual methods for calculating health indicator are still the most popular and common ones, because such selected indicator has an obvious trend to make it easier to predict the RUL of bearing. But this still requires much expert knowledge on mechanical system, degradation theory and statistics. Besides, data-driven methods have always been employed in the second step to regress the degradation curve of health indicator, but the failure point is still calculated according to an empirical special threshold in general.

Furthermore, accurate and effective prognosis in mechanic system is still hampered by a vital drawback of RNN. The collected data responding to the whole life of the machine is so long that RNN failed to process it, due to the so-called vanishing and exploding gradient problems [30]. To the authors' best knowledge, most of proposed RNN-based models for prognosis make do with sliced data instead of the whole long data [31], which may cause some problems such as overfitting and decrease the prognosis accuracy.

In order to solve the aforementioned shortcoming, this paper proposes an RNN model based on encoder–decoder structure with attention mechanism. First, five band-pass energy values of frequency spectrum of vertical and horizontal vibration signals are used as input to train and test the proposed RNN model. Different from traditional RNN structure, attention machine is added to decide the attention distribution according to the first look by encoder, which help decoder make a better prognosis and overcome the vanishing and exploding gradient problems. As a result, a sequence of HI values ranges from 0 to 1 is supposed to be obtained. Finally, by linear regression with least square method, the accurate prediction of bearing RUL can be calculated. The main contributions of this paper are summarized as follows.

(1) A novel method with little prior knowledge for rolling bearing RUL prediction is proposed, and achieved high prognosis accuracy.

(2) An RNN model based on encoder–decoder structure with attention mechanism is proposed to mine useful degradation information from long historic data, which is validated by visualizing the attention distribution.

The rest of this paper is arranged as follows. Section 2 introduces the basic theories of RNN, LSTM, GRU (Gated Recurrent Unit), encoder–decoder structure and attention mechanism. The detailed process of proposed approach is then described in Section 3. In Section 4, our method is validated using the dataset from the accelerated degradation testing on rolling element bearings. Furthermore, the analysis of the proposed neural network is also presented to explain its effectiveness. Finally, conclusions are drawn in Section 5.

2. Theoretical background

To deal with prognosis problem in bearing RUL, it is essential to obtain degradation tendency according to historical information with uncertain length. And RNN is designed to process such sequence data. In this section, some theories about RNN, encoder–decoder frameworks and attention mechanism are introduced in brief.

2.1. RNN

RNNs [32] are a family of neural networks for processing variable-length sequential data by having a recurrent hidden state whose activation at each time is dependent on that of the previous time. Formally, given a sequence data $\{x_k\}$, recurrent neuron updates its hidden state h_t and outputs $\hat{y}_t$ at time t as follow:

$$h_t = \tanh(\boldsymbol{U}x_t + \boldsymbol{W}h_{t-1} + b) \tag{1}$$

$$\hat{y}_t = f(\boldsymbol{V}h_t + c) \tag{2}$$

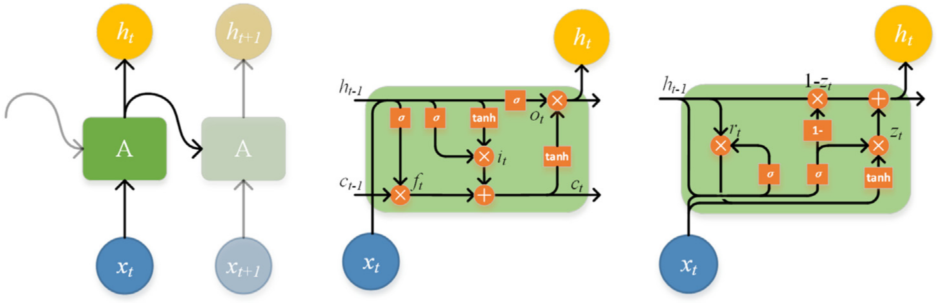
where $\boldsymbol{U}$, $\boldsymbol{W}$ and $\boldsymbol{V}$ are the training weights respectively for input-to-hidden, hidden-to-hidden and hidden-to-output connections, while b and c are the bias vectors which allow each node to learn an offset. Normally, the activation function in the process of obtaining current hidden state h_t is $\tanh(\cdot)$, to keep hidden state in the range of $[-1, 1]$. And the activation function of output, $f(\cdot)$, is decided by different task. Consequently, the current hidden state h_t contains all the previous information, and the output $\hat{y}_t$ is influenced by both current input information and the historic ones.

As shown in Fig. 1(a), current network goes across a sequence step by step. Unfortunately, it has been observed by, e.g., Bengio et al. [30] that it is difficult to train RNN model to capture long-term dependencies because the gradients tend to either vanish or explode. In order to address this challenging problem, new recurrent network architectures like LSTM and GRU are proposed [33].

Instead of a simple hidden neuron, LSTM has a separate memory cell and three gates to control the information flow, as described in Fig. 1(b). Described by following formulas, forget gate f_t, input gate i_t and output gate o_t are calculated according to current input x_t and last hidden state h_{t-1}. In each step, current separate cell C_t updates by forgetting some information and adding some information from current input x_t and last hidden state h_{t-1}, which is decided by forget gate f_t and input gate i_t respectively. While the current hidden state h_t is decided by present memory cell C_t and output gate o_t.

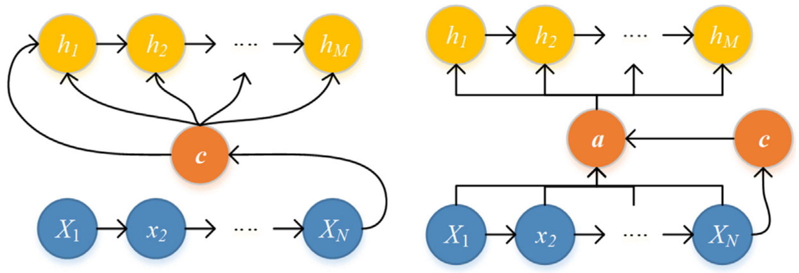
$$f_t = \sigma(\boldsymbol{U_f}x_t + \boldsymbol{W_f}h_{t-1} + b_f) \tag{3}$$

$$i_t = \sigma(\boldsymbol{U_i}x_t + \boldsymbol{W_i}h_{t-1} + b_i) \tag{4}$$



(a) RNN (b) LSTM (c) GRU

Fig. 1. Well-known recurrent neural networks: (a) RNN; (b) LSTM; (c) GRU.



(a) Encoder-decoder (b) Attention mechanism

Fig. 2. Encoder–decoder framework and attention mechanism.

$$o_t = \sigma(\boldsymbol{U_o}x_t + \boldsymbol{W_o}h_{t-1} + b_o) \quad (5)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(\boldsymbol{U_C}x_t + \boldsymbol{W_C}h_{t-1} + b_C) \quad (6)$$

$$h_t = o_t \odot \tanh C_t \quad (7)$$

In formulas (3)–(5), $\boldsymbol{U_f}(\boldsymbol{U_i}\backslash\boldsymbol{U_o})$, $\boldsymbol{W_f}(\boldsymbol{W_i}\backslash\boldsymbol{W_o})$ and $b_f(b_i\backslash b_o)$ are training matrix weights and offset, which learn the relationship among current input x_t, last output h_{t-1} and forget (input\output) gate. And $\sigma(\cdot)$ is a nonlinear activation function that restricts the gate value into a particular interval (usually [0, 1]). Then, in formula (6), the output of $\tanh(\boldsymbol{U_C}x_t+\boldsymbol{W_C}h_{t-1}+b_C)$ with training matrix weights, $\boldsymbol{U_C}$ and $\boldsymbol{W_C}$, and offset b_C, is a candidate value regulated by input gate i_t that will be added to the new state of memory cell C_t, together with the value of old state C_{t-1} regulated by forget gate f_t, where $\odot$ is the point wise multiplication operator. Eventually, formula (7) represents the update process of final hidden state h_t based on memory cell C_t.

Unlike to the traditional recurrent unit which overwrites its content at each time step, LSTM unit is capable to decide whether to keep the existing memory via three gates. Consequently, important information from early stage can be easily carried over a long distance while the useless ones would be abandoned, which means LSTM is easier to capture potential long-distance dependencies.

Similar to the LSTM unit, GRU shown in Fig. 1(c), proposed by Cho et al. also has gating units that modulate the information flow. However, without separate memory cell, GRU can be regarded as a simplification of LSTM unit. Mathematically, the computation of GRU can be described as follows:

$$r_t = \sigma(\boldsymbol{U_r}x_t + \boldsymbol{W_r}h_{t-1} + b_r) \quad (8)$$

$$z_t = \sigma(\boldsymbol{U_z}x_t + \boldsymbol{W_z}h_{t-1} + b_z) \quad (9)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tanh(\boldsymbol{U_C}x_t + r_t \odot (\boldsymbol{W_C}h_{t-1}) + b_C) \quad (10)$$

where r_t and z_t are the reset gate and update gate with respective training weights and offsets. The update process of hidden state h_t at time t has been described in Eq. (10). The update gate z_t decides how much the unit updates its activation, or content, while the reset gate r_t decides how much of the previous state should be forgotten.

The most prominent change of LSTM and GRU is the additive gates that allow neurons select and maintain important information for a long series of steps. Furthermore, this addition gate also creates shortcut paths that allow the error to be back-propagated easily without too quickly vanishing. And more and more experiments [34] prove that, LSTM and GRU could with sequence with length over one hundred, while traditional RNN only handles with dozens of steps.

2.2. Encoder–decoder framework and attention mechanism

RNN Encoder–Decoder is proposed by Cho et al. [35] for statistical machine translation. This neural network framework consists of two components, an encoder which computes a representation $\boldsymbol{c}$ for an input sequence $\{x_N\}$ and a decoder which generates one target sequence step by step, as illustrated in Fig. 2(a). Therefore, at step t, the conditional distribution of y_t is:

$$\mathrm{P}\,(y_t|y_{t-1}, y_{t-2}, \ldots, y_1, \boldsymbol{c}) = g(h_t, y_{t-1}, \boldsymbol{c}) \quad (11)$$

where h_t is the hidden state at step t and $g(\cdot)$ is an activation function, such as softmax, which produces valid probabilities. [35]

Generally, both encoder and decoder are RNN models, which can deal with variable-length sequence. The last hidden state of RNN encoder not only acts as a representation $\boldsymbol{c}$ for input sequence, but is also passed as the initial hidden state of RNN decoder. In this way, output y_t at any time t is decided by both the whole input $\{x_k\}$ and the last output y_{t-1}, so as to ensure the accuracy and continuity for each y_t. Furthermore, encoder–decoder framework allows to produce output sequence with different length from input sequence, which makes it more flexible.

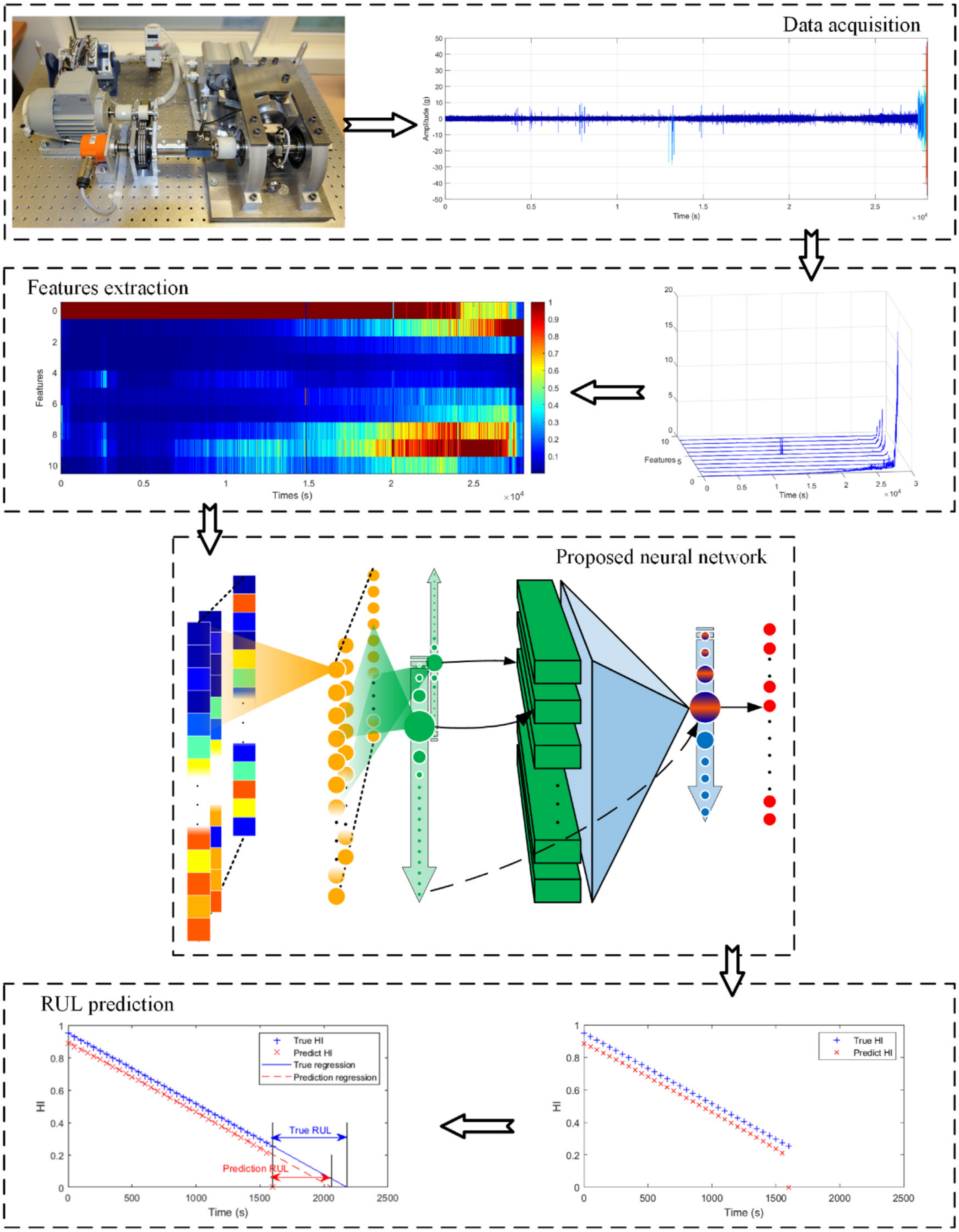


Fig. 3. A flowchart of the proposed method.

However, due to the shortcoming of RNN in dealing with very long sequence as mentioned above, the last hidden state of RNN encoder only carries the most vital information, which is still too insufficient to mine underlying information in input sequence, in the case that the input sequence is very long. Many researchers suggest to make use of the output of RNN encoder [36,37], and the attention machine [38] is proved to be a wise choice. As displayed in Fig. 2(b), attention machine acts as a gate function, producing a more flexible gate a_t with length as input hidden state, to choose the most vital part of the input in each time step. Instead of passing the last hidden state of RNN encoder directly as the representation, the point multiplication of RNN encoder output and attention gate as a more flexible representation makes contribution to decide the output possibility of y_t, as in Eqs. (12)–(14). formally. At each step, attention gate a_t is calculated according to the scores distribution which is related with the current target hidden state h_t and each encoder hidden state $\overline{h_s}$ (, whose collection is denoted as $\overline{h}$).

$$\mathrm{P}\left(y_t | y_{t-1}, y_{t-2}, \ldots, y_1, \mathbf{c}\right) = g(h_t, y_{t-1}, (a_t \cdot \overline{h})) \tag{12}$$

$$a_t\left(s\right) = \mathrm{align}\left(h_t, \overline{h_s}\right) = \frac{\exp(\mathrm{score}(h_t, \overline{h_s}))}{\sum_{s'} \exp(\mathrm{score}(h_t, \overline{h_{s'}}))} \tag{13}$$

$$\text{score}\left(h_t, \overline{h_s}\right) = \begin{cases} h_t^{\mathrm{T}}\overline{h_s} & \textit{dot} \\ h_t^{\mathrm{T}}\boldsymbol{W_a}\overline{h_s} & \textit{general} \\ v_a^{\mathrm{T}}\tanh(\boldsymbol{W_a}[h_t;\overline{h_s}]) & \textit{concat} \end{cases} \tag{14}$$

where $\boldsymbol{W_a}$ and v_a are trainable weight and vector that allow attention machine to learn some knowledge between the importance of input and every output.

Besides, the visualization of attention attribution is more explicable than the weights of neural units. For example, in the field of neural machine translation, visualization of attention weight is able to point out the relation between words from source language and target language [38].

3. Prognostic procedure

In this section, the procedure of the proposed approaches is described in detail. As shown in Fig. 3, this process mainly contains three steps, feature extraction, HI calculation by RNN and bearing RUL prediction via linear regression. In the first step, energy values of five sub-bands frequency spectra are extracted from raw signal as the features. Then, a sequence of HI values between 0 and 1 is obtained as the output of the proposed neural network with features as input. Finally, the RUL value in the end of the collected signal can be calculated via scaling up the last HI value with linear regression. Detail information is given as follow.

3.1. Feature extraction and normalization

In this paper, energy values of five sub-bands frequency spectra are extracted from raw signal as the features. Supposed that the highest frequency of frequency spectrum is 25.6 kHz, and then the five sub-bands frequency spectra locate in 0–5.12 kHz, 5.12–10.24 kHz, 10.24–15.36 kHz, 15.36–20.48 kHz and 20.48–25.6 kHz respectively, as displayed in Eq. (15). Then, features are normalized into interval [0, 1], as shown in Eq. (16).

$$E_{t,i} = \frac{1}{5120}\int_{i\times 5120}^{(i+1)\times 5120} X_t(f)df \tag{15}$$

$$\widetilde{E_{t,i}} = \frac{E_{t,i} - \min_{0\le i<5}(E_t)}{\max_{0\le i<5}(E_t) - \min_{0\le i<5}(E_t)} \tag{16}$$

where $E_{t,i}$ is the ith feature in time step t, $X_t(f)$ is the frequency of vibration signal in time step t. And after normalization, the features $\widetilde{E_{t,i}}$ as the input of the proposed neural network are obtained. Finally, 10 feature values are obtained from both vertical and horizontal vibration signals. And the total time series features can be visualized as line chart in Fig. 4.

Compared with statistical features from time domain, such features are more independent to some extent. Moreover, normalization (or fusing) methods are essential for statistical features from time domain, where an inappropriate one may cause difficulties to train a successful data-driven model. But such problems would not happen when dealing with frequency features, for the reason that the variation intervals of frequency spectrum would not change a lot.

3.2. Health indicator calculation via neural network

HI is a custom indicator that helps researchers to estimate the health states of machinery in real time [15]. In RUL prognosis, HI with ability to reveal the degradation processes is a significant tool, but still lacking a common and suitable resolution. Besides, an empirical threshold is required in RUL prediction though the HI value is given. As a result, a lot of prior knowledge is essential to construct an appropriate HI in previous related works, which against the original intention of using data-driven methods.

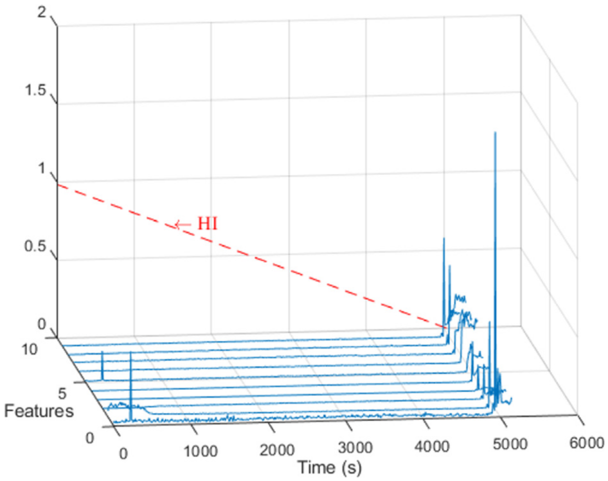


Fig. 4. Comparison between features and HI curve.

In order to reduce such manual and empirical process, or reduce the demand of expensive prior expert knowledge, a simple HI is proposed according to the responding RUL. Formally, for an existing bearing degradation signal, the HI at time t can be obtained by respective RUL divided by the RUL in origin (RUL_0):

$$\mathrm{HI}_t = \frac{\mathrm{RUL}_t}{\mathrm{RUL}_0} \tag{17}$$

Then, a sequence of HI values located in interval [0, 1] is acquired, and the failure point is the one when HI reaches 0 (RUL_t goes to 0). The curves of extracted features and the responding HI are displayed in Fig. 4, which are the input and the output of neural network. The curve of HI is a simple straight line, whose value changes from 1 to 0 as time goes. In this way, HI would be easier to predict, regress and calculate the RUL.

As shown in Fig. 4, few relations between features and HI values can be found intuitively. In order to mine such underlying relation and take advantages of mutual features, an RNN with encoder–decoder framework and attention mechanism is proposed.

The whole structure of the proposed network is illustrated in Fig. 5. First, a convolution layer acts as a feature compressor, so that the total length of sequences would be reduced to an appropriate size, which relieves the impact of gradient vanishing or exploding in RNN models. Then a bidirectional GRU network encoder is used to take a first look of the features sequence, and output the hidden state h_t. Finally, another GRU network acts as decoder to predict the HI values step by step. In each step, attention score according to the presentation of the whole information provided by encoder helps to find out the most important information.

In order to get a better consequence, train data should be similar with test data. And in PHM dataset, test data does not go to the end. Therefore, in training stage, instead of the whole degradation signal, a random length signal from the origin point to a nearly end point is used as the training sequence, as a simulation of test signals, which is shown in Fig. 6.

And mean square error is used as the cost function to train the proposed network:

$$L = \frac{1}{N}\sum_{t=0}^{N}(y_t - \widehat{y_t})^2 \tag{18}$$

where y_t and $\widehat{y_t}$ are the true HI values and predicted HI values. By minimizing the mean square error, the curve of predicted HI

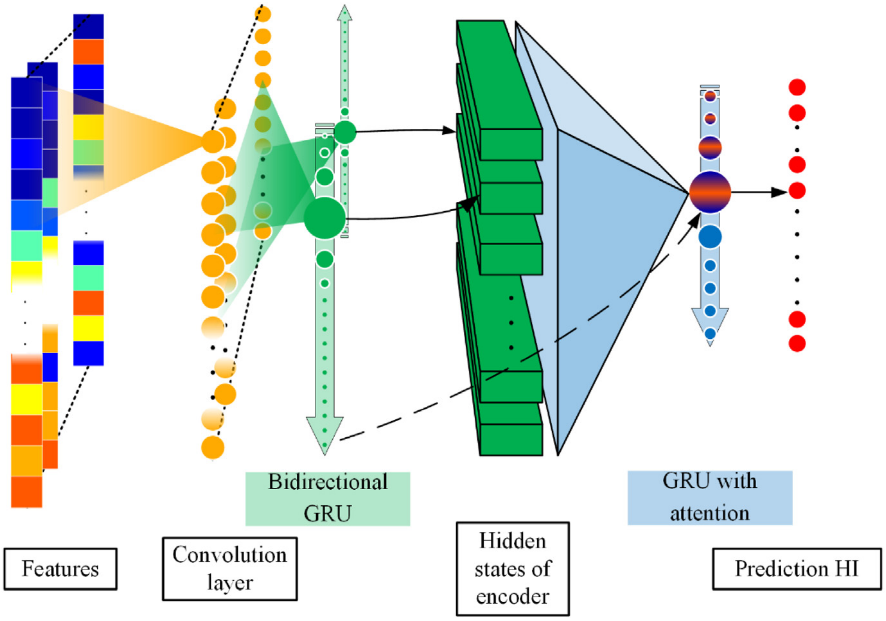


Fig. 5. Overview of the proposed neural network.

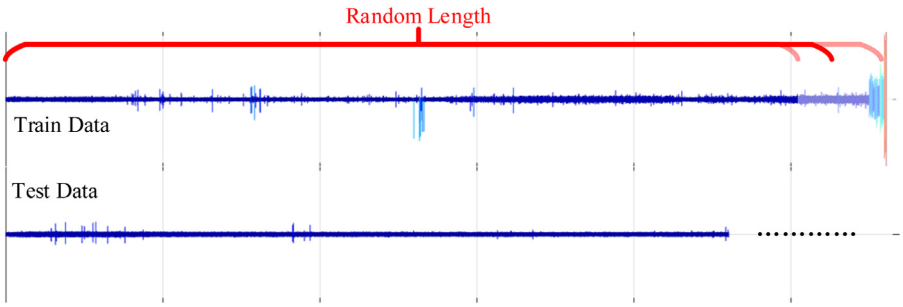


Fig. 6. Training data with random length.

values is expected to linear and monotonic which makes it easy to be linear regressed and calculated the tendency.

Besides, in order to obtain a good model, some training tricks have been applied. As advised by Rafal Jozefowicz, et al. [39], the bias of reset gate of bidirectional GRU in encoder is initialized as one to help the network learning the time-dependent relation. And Adam algorithm [40] is used as the optimizer with learning rate setting as 0.005. The training and testing processes are realized in python 3.5 with pytorch 0.4.1, based on hardware platform with CPU as i3-8100 and GPU as GTX 1070. Formally, these hyper parameters are listed in Table 1.

3.3. RUL calculation via linear regression

After HI values are predicted, bearing RUL can be calculated easily by scaling up the last HI values, where the scaling factor is obtained via linear regression, as shown as the last step in Fig. 3. Formally, a linear equation $y = at + b$ is used to linear fitting all the predicted HI values, where a and b are the parameters need to be estimated. According to the least square method, the parameters a and b can be acquired via Eqs. (19) and (20):

$$a = \frac{\sum_{i=0}^{N} t_i y_i - (N+1)\bar{t}\bar{y}}{\sum_{i=0}^{N} t_i^2 - (N+1)\bar{t}^2} \tag{19}$$

$$b = \bar{y} - a\bar{t} \tag{20}$$

Then, the predicted RUL ($\widehat{\mathrm{RUL}}$) can be calculated as follow:

$$\widehat{\mathrm{RUL}} = \frac{b}{a} - t_N \tag{21}$$

where t_i and y_i are record time and respective predict HI values in ith step, and t_N is the record time in the final (Nth) step.

4. Experiment verification

In order to validate the proposed method, an experimental dataset is introduced as training and testing data to evaluate its performance in this section. Further comparison with recent novel methods is also carried in the later part. And some analyzes are carried out in final.

4.1. Dataset description

In this paper, validation dataset is from PRONOSTIA collected by conducting accelerated bearing degradation test rig, as shown in Fig. 7. Both horizontal and vertical vibration signals are monitored by accelerometers adhered on the external race of the bearing. When the test rig starts, per record of the measure vibration signals, which lasted 0.1 s with a sampling frequency 25.6 kHz, are recorded every 10 s. More detailed information about the platform and experiments can be found in [41].

In the experiments, three different working conditions were considered: the first ones operated at a rotation speed as 1800

Table 1
Hyper parameters of the proposed neural network.

	Parameters
Convolution layer	Kernel size: 8; Strikes: 5; Filters: 64; Activate function: PReLu
Bidirectional GRU	Hidden size: 200; Bias of reset gate initialized as 1
Decoder GRU	Hidden size: 200; Attention calculation method: concat
Loss function	Mean square error (MSE)
Optimizer	Adam with learning rate as 0.005

Table 2
Datasets of IEEE 2012 PHM Prognostic Challenge.

Operating conditions	Conditions 1	Conditions 2	Condition 3
Rotation speed	1800 rpm	1650 rpm	1500 rpm
Loading force	4000 N	4200 N	5000 N
Learning set	Bearing1_1 Bearing1_2	Bearing2_1 Bearing2_2	Bearing3_1 Bearing3_2
Test set	Bearing1_3 Bearing1_4 Bearing1_5 Bearing1_6 Bearing1_7	Bearing2_3 Bearing2_4 Bearing2_5 Bearing2_6 Bearing2_7	Bearing3_3

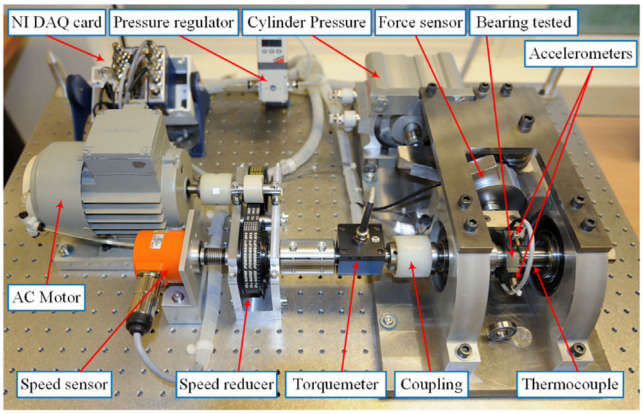


Fig. 7. Overview of test rig of PRONOSTIA.

rpm with loading force as 4000 N, while the second ones at 1650 rpm with 4200 N and the third ones at 1500 rpm with 5000 N. In each working condition, several bearings are tested, specially, 7 bearings are tested in the first condition, 7 in the second one and 3 in the third one with name from bearing1_1 to bearing1_7, from bearing2_1 to beairng2_7 and from bearing3_1 to bearing 3_3 respectively. As a dataset for PHM challenge, only data from first two bearings in each condition is complete (keep measuring to the end), while the others as testing data lack vibration signals in the last stage (keep measuring until the omen comes). All this information is listed in Table 2. Fig. 8 shows the vibration signal of bearing1_1 during its whole life cycle. And it is intuitive that the amplitudes increase over time, especially in last stage, which proves that the degradation tendency can be revealed from vibration signal.

4.2. Evaluation index

To make a comparison with the performance of prognosis methods, appropriate evaluation indexes are required. In the IEEE PHM 2012 challenge [41], a score function is used to assess such estimation approaches. This score can be calculated following equation (22)–(24).

$$Er_i = 100 \times \frac{ActRUL_i - \widehat{RUL_i}}{ActRUL_i} \tag{22}$$

$$A_i = \begin{cases} \exp(-\ln(0.5)\cdot(Er_i/5)) & \text{if } Er_i \le 0 \\ \exp(+\ln(0.5)\cdot(Er_i/20)) & \text{if } Er_i > 0 \end{cases} \tag{23}$$

$$Score = \frac{1}{N}\sum_{i=1}^{N} A_i \tag{24}$$

Let note $\widehat{RUL_i}$ and $ActRUL_i$ respectively the remaining useful life of the ith bearing estimated by a prognosis method, and the actual RUL to be predicted. Then, Er_i is respective percent errors, according to which A_i as an evaluation index in each prognosis case can be obtained. And $Score$ is the average value of A_i. Fig. 9 depicts the evolution of scoring function.

As shown in Fig. 9, the score will be higher when Er_i is positive, which means that an early prediction is more valuable than a late one. This score function is very fair considered the practicability, but fail to evaluate the accuracy of predictions. In order to evaluate the accuracy of predictions, the average value of Er_i, which is used in [32], is also considered in this paper. Furthermore, the precision of the predictions reflects the stability of prognosis methods, which is measured by the absolute mean of Er_i in this paper. And such two scores, described as $\overline{Er}$ and $\overline{|Er|}$, can be obtained as follow:

$$\overline{Er} = \frac{1}{N}\sum_{i=1}^{N} Er_i \tag{25}$$

$$\overline{|Er|} = \frac{1}{N}\sum_{i=1}^{N} |Er_i| \tag{26}$$

4.3. RUL prediction and discussion

After training the proposed network with training data, the RUL of each test bearing in dataset can be obtained through the procedure mentioned in Section 3. And the predict RUL and respective error are displayed in Fig. 10. In Fig. 10, the bevels of blue and red semitransparent right-angled trapezia represent the true HI values and the regressive line of predict HI values. Then, extension lines are drawn, whose crossover points in plane XOY decide the prediction error. For distinguishing more easily, the prediction errors of early prediction or late prediction are marked in green or red bar, respectively. It is clarified that little prediction errors happened, except in the cases of Bearing1_7 and Bearing2_3, which reveals the good RUL prognosis ability of our methods. Divided by the actual RUL, percent error Er is shown in Fig. 11, as a bar chart in plane XOY. Since the actual RUL of Bearing1_4 and Bearing1_5 is very low, divided by which high percent errors come out. Besides, scores of each dataset are also calculated according to Eq. (23) and displayed as solid bars in Fig. 11. Except such 4 cases (Beaing1_4, Bearing1_5, Bearing1_7 and Bearing2_3), the other percent errors and responding scores, as illustrated prisms in Fig. 11, perform very well.

Furthermore, such results and the evaluation indexes mentioned in Section 4.2 are compared with similar works from [32, 42–45]. This is listed in Table 3, where the results in the 4-th column are ours. In Table 3, the values in the column of current time represent how much time the measuring process is keeping, which also means how much data is available to calculate the

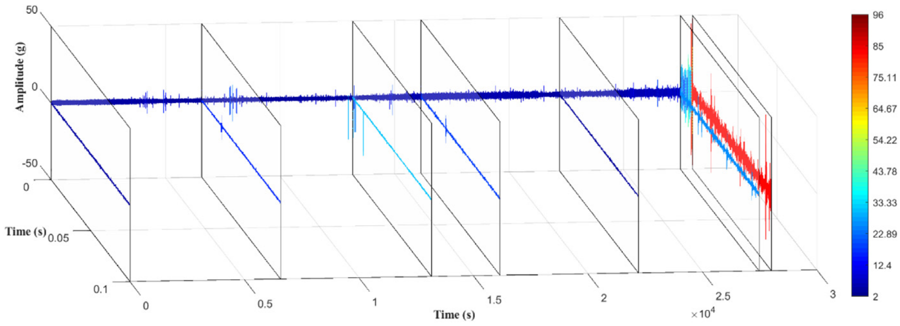


Fig. 8. Vibration signals of Bearing1_1.

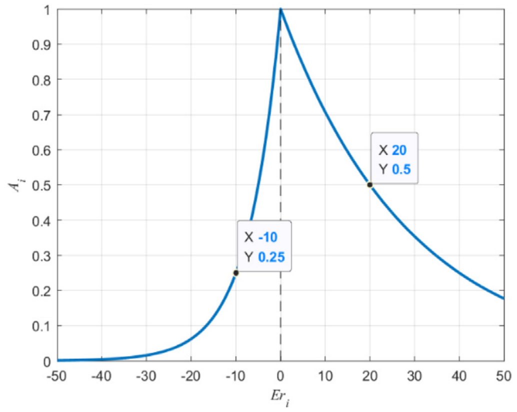


Fig. 9. Scoring function of a RUL estimates according to its percent error.

RUL. From Table 3, the proposed approach has the lowest percent error and the highest score than the other ones, but with a high absolute percent error. Namely, our approach gives the most accurate RUL prediction, which is more valuable in practice, but the precision is still required to be improved.

For a further exploring to the inner mechanism of the proposed neural network, visualization of the attention attribution in the cases of test dataset is shown in Fig. 12, where HS represents the hidden state of encoder. In this visualization, the hotter the color is, the more attention is distributed. Generally, more attention is located in the beginning of the hidden state of encoder, and less change can be found along HI time-axis. Such phenomenon is supposed to be the result of predicting a linear HI function. Since the predict HI is expected to be linear, little information is required to give an accuracy prediction. Namely, the slope of the true HI values, which is closely related with the degradation trend of the bearing, is the most vital parameter. And the neural network learns that to calculate HI values according to the slope is an easier and more accurate way. In order to obtain this slope, comparison between the final state and the beginning state of the input data is essential. Besides, in the bidirectional GRU layer, memory units would record the most information of the input data when going in both directions, theoretically. While in actual data shown in Fig. 4, violent changes only happened in the last stage, which carried the information of the prognostic of bearing faults and should be caught by memory units. As a result, the memory unit started from the starting point would capture such information finally, but lost the information of the starting point during its update procedure. On the other hand, the memory unit started from the ending point not only keeps such information but also have an overview of the information of the beginning state, whose information is saved in the beginning of the hidden stage of the encoder output. Therefore, the beginning of the hidden state of the encoder output attracts the most attention and attention attribution does not change a lot along the axis of HI time.

4.4. RUL prediction in early stage

In this section, to evaluate the performance of the proposed method in actual industry situation, experiment about RUL prediction in early stage is conducted. For each test dataset, 20%, 40%, 60%, 80% and 100% of the whole dataset is put into the neural network. The predict HI and responding predict RUL values of dataset Bearing1_4 and Bearing3_3 are shown in Fig. 13. Prediction results with different lengths of sequence data are named

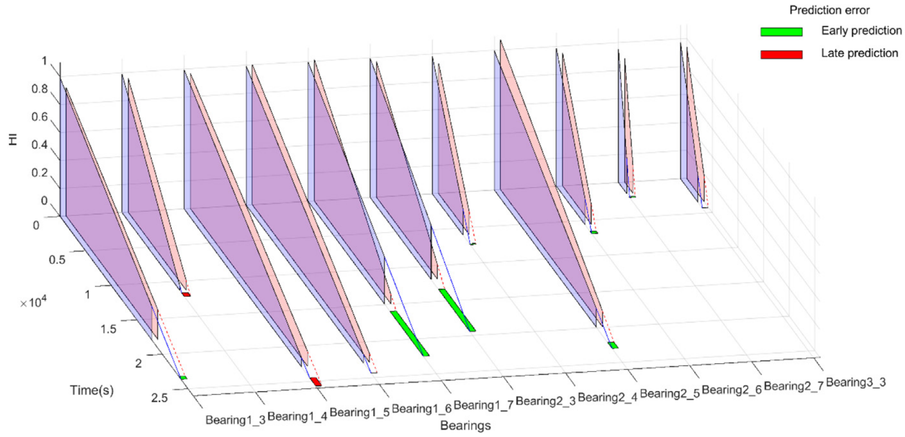


Fig. 10. Overview of prediction error.

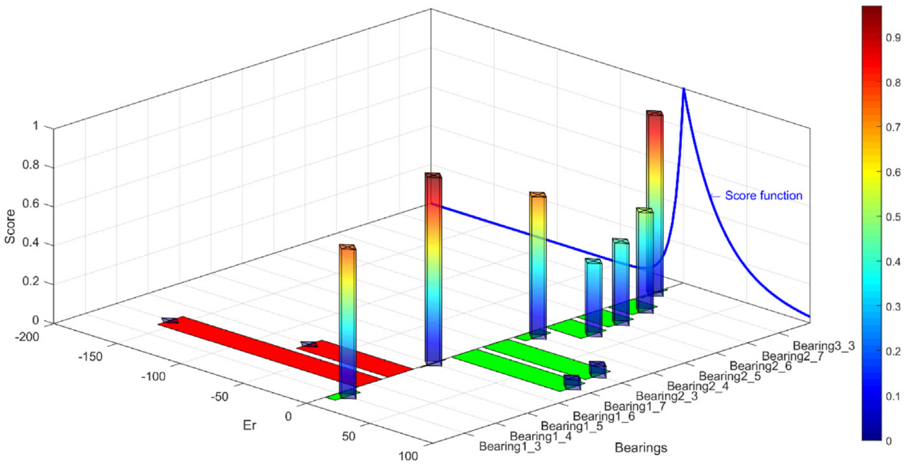


Fig. 11. Overview of Er_i and A_i.

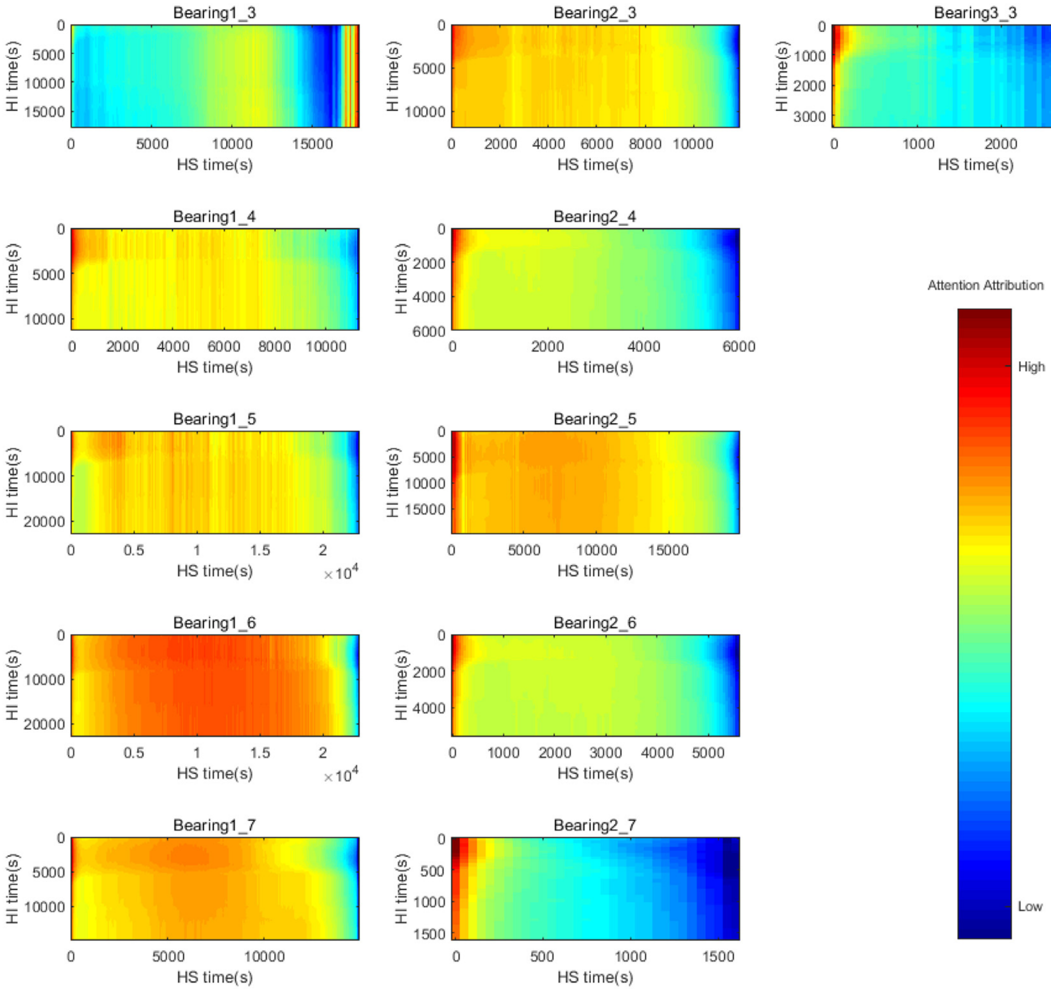


Fig. 12. Attention attribution of test dataset.

as the 1st, 2nd, 3rd, 4th and 5th prediction. And the predict RUL values are fitting with a 4-order polynomial curve.

From the prognosis results in Fig. 13, the proposed approach tends to give a low HI and a very early prediction RUL in early stage, when the prognostic of bearing faults did not happen. But when the prognostic happens, in the case of 4th and 5th prediction, the predict HI values are closed to the true ones and the predict RUL curve also tends to be the same trend with the true RUL. To summarize, the predict RUL values in early stage are unfortunately useless. But the curve of predict RUL values also points out that, when it goes downs, the prognostic of bearing

Table 3
RUL prediction results and comparison.

Testing dataset	Current time (s)	Actually RUL (s)	Predict RUL (s)	*Er*	*Er* [32]	*Er* [42]	*Er* [43]	*Er* [44]	*Er* [45]
Bearing1_3	18010	5730	5293.5	7.62	43.28	37	54.73	−0.35	−1.04
Bearing1_4	11380	339	873.6	−157.71	67.55	80	38.69	5.6	−20.94
Bearing1_5	23010	1610	2778.39	−72.57	−22.98	9	−99.4	100	−278.26
Bearing1_6	23010	1460	1446.4	0.93	21.23	−5	−120.07	28.08	19.18
Bearing1_7	15010	7570	1060.6	85.99	17.83	−2	70.65	−19.55	−7.13
Bearing2_3	12010	7530	1412.4	81.24	37.84	64	75.53	−20.19	10.49
Bearing2_4	6110	1390	1264.4	9.04	−19.42	10	19.81	8.63	51.8
Bearing2_5	20010	3090	2218.9	28.19	54.37	−440	8.2	23.3	28.8
Bearing2_6	5710	1290	968.5	24.92	−13.95	49	17.87	58.91	−20.93
Bearing2_7	1710	580	469.5	19.06	−55.17	−317	1.69	5.17	44.83
Bearing3_3	3510	820	802.9	2.09	3.66	90	2.93	40.24	−3.66
$\overline{Er}$				**2.62**	12.20	−38.64	6.42	20.89	−16.08
$\overline{\lvert Er\rvert}$				44.49	32.48	100.27	46.32	**28.18**	44.28
Score				**0.4384**	0.2631	0.3066	0.3829	0.4285	0.3550

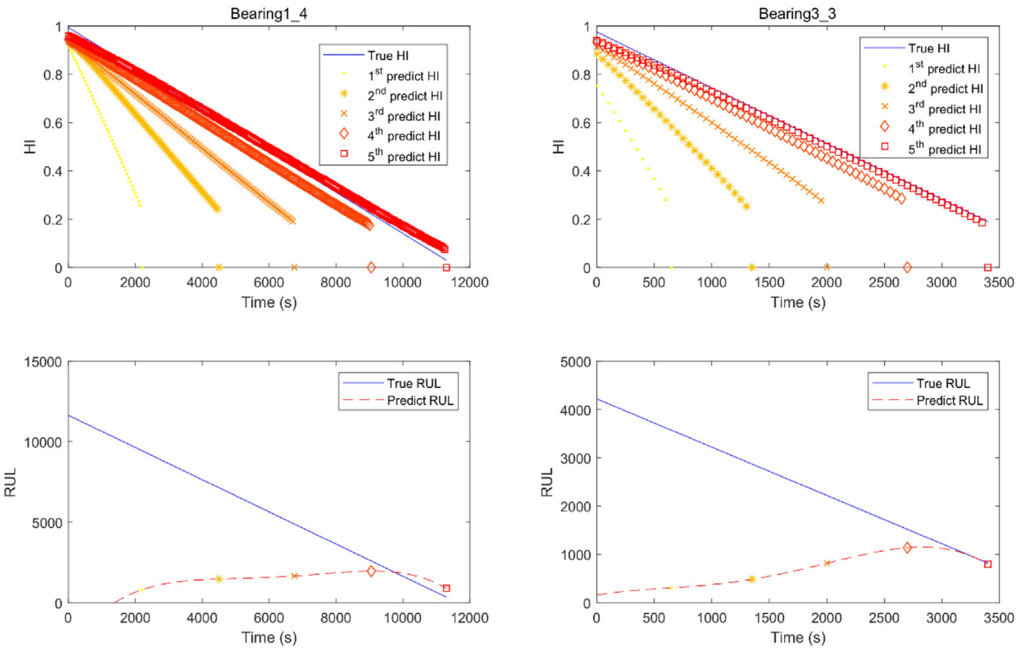


Fig. 13. RUL prediction of Bearing1_4 and Bearing3_3 in early stage.

faults may have happened and the predict RUL values are worth considering.

5. Conclusion

Accurate RUL prediction highly depends on using the long-time-depended information from the long-time sequence data effectively. In this paper, a pure data-driven approach based on encoder–decoder framework is proposed. During the proposed procedure of RUL prediction, features extraction and threshold setting is automatic and without any prior expert knowledge. Finally, in the validation, the proposed method achieves the lowest average percent error and highest average score compared with other novel methods. Moreover, experiments on RUL prediction in early stage also prove the difficulties of accurate prognosis before an omen happens. But the prediction of the proposed method in the final stage is proved to be valuable.

There are still many shortcomings in the proposed method, such as low precision and crude approaches for automatic feature extraction and HI construction. In future, more efforts are required to deal with the problem of intelligent feature extraction and HI construction to build up a complete end-to-end deep learning method.

Funding

This work was supported by the National Natural Science Foundation of China (Grant no. 51875138), and the National Science and Technology Major Project of China (863 Program, No. 2016ZX04002-004).

References

[1] Y. Lei, J. Lin, M.J. Zuo, Z. He, Condition monitoring and fault diagnosis of planetary gearboxes: A review, Measurement 48 (2014) 292–305.

[2] A. Ghods, H.H. Lee, Probabilistic frequency-domain discrete wavelet transform for better detection of bearing faults in induction motors, Neurocomputing 188 (2016) 206–216.

[3] J. Liu, W. Wang, F. Golnaraghi, An enhanced diagnostic scheme for bearing condition monitoring, IEEE Trans. Instrum. Meas. 59 (2) (2010) 309–321.

[4] Y. Wang, G. Xu, Q. Zhang, D. Liu, K. Jiang, Rotating speed isolation and its application to rolling element bearing fault diagnosis under large speed variation conditions, J. Sound Vib. 348 (2015) 381–396.

[5] F. Jia, Y. Lei, J. Lin, X. Zhou, N. Lu, Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, Mech. Syst. Signal Process. 72 (2016) 303–315.

[6] J. Zhu, N. Chen, W. Peng, Estimation of bearing remaining useful life based on multiscale convolutional neural network, IEEE Trans. Ind. Electron. 66 (4) (2019) 3208–3216.

[7] M. Jouin, R. Gouriveau, D. Hissel, M.C. Péra, N. Zerhouni, Particle filter-based prognostics: Review, discussion and perspectives, Mech. Syst. Signal Process. 72 (2016) 2–31.

[8] M. Jouin, R. Gouriveau, D. Hissel, M.C. Péra, N. Zerhouni, Degradations analysis and aging modeling for health assessment and prognostics of PEMFC, Reliab. Eng. Syst. Saf. 148 (2016) 78–95.

[9] J.B. Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, F. Fnaiech, Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network, Mech. Syst. Signal Process. 56 (2015) 150–172.

[10] X.S. Si, W. Wang, C.H. Hu, D.H. Zhou, Remaining useful life estimation–a review on the statistical data driven approaches, European J. Oper. Res. 213 (1) (2011) 1–14.

[11] L. Ren, J. Cui, Y. Sun, X. Cheng, Multi-bearing remaining useful life collaborative prediction: A deep learning approach, J. Manuf. Syst. 43 (2017) 248–256.

[12] R. Huang, L. Xi, X. Li, C.R. Liu, H. Qiu, J. Lee, Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods, Mech. Syst. Signal Process. 21 (1) (2007) 193–207.

[13] C. Chen, B. Zhang, G. Vachtsevanos, M. Orchard, Machine condition prediction based on adaptive neuro–fuzzy and high-order particle filtering, IEEE Trans. Ind. Electron. 58 (9) (2011) 4353–4364.

[14] T.H. Loutas, D. Roulias, G. Georgoulas, Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression, IEEE Trans. Reliab. 62 (4) (2013) 821–832.

[15] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: A systematic review from data acquisition to RUL prediction, Mech. Syst. Signal Process. 104 (2018) 799–834.

[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[17] G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 30–42.

[18] N. Boulanger-Lewandowski, Y. Bengio, P. Vincent, Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription, 2012, arXiv preprint arXiv:1206.6392.

[19] Z. Huijie, R. Ting, W. Xinqing, Z. You, F. Husheng, Fault diagnosis of hydraulic pump based on stacked autoencoders, in: Electronic Measurement & Instruments, ICEMI, 2015 12th IEEE International Conference on, Vol. 1, IEEE, 2015, pp. 58–62.

[20] L. Guo, H. Gao, H. Huang, X. He, S. Li, Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring, Shock Vib. 2016 (2016).

[21] N.K. Verma, V.K. Gupta, M. Sharma, R.K. Sevakula, Intelligent condition based monitoring of rotating machines using sparse auto-encoders, in: Prognostics and Health Management, PHM, 2013 IEEE Conference on, IEEE, 2013, pp. 1–7.

[22] W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load, Mech. Syst. Signal Process. 100 (2018) 439–453.

[23] W. Zhang, G. Peng, C. Li, Y. Chen, Z. Zhang, A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals, Sensors 17 (2) (2017) 425.

[24] Y. Chen, G. Peng, C. Xie, W. Zhang, C. Li, S. Liu, ACDIN: Bridging the gap between artificial and real bearing damages for bearing fault diagnosis, Neurocomputing 294 (2018) 61–71.

[25] C. Li, W. Zhang, G. Peng, S. Liu, Bearing fault diagnosis using fully-connected winner-take-all autoencoder, IEEE Access 6 (2018) 6103–6115.

[26] Z. Zhu, G. Peng, Y. Chen, H. Gao, A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis, Neurocomputing 323 (2019) 62–75.

[27] A. Malhi, R. Yan, R.X. Gao, Prognosis of defect propagation based on recurrent neural networks, IEEE Trans. Instrum. Meas. 60 (3) (2011) 703–711.

[28] M. Yuan, Y. Wu, L. Lin, Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network, in: Aircraft Utility Systems, AUS, IEEE International Conference on, IEEE, 2016, pp. 135–140.

[29] R. Zhao, J. Wang, R. Yan, K. Mao, Machine health monitoring with LSTM networks, in: Sensing Technology, ICST, 2016 10th International Conference on, IEEE, 2016, pp. 1–6.

[30] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (2) (1994) 157–166.

[31] L. Guo, N. Li, F. Jia, Y. Lei, J. Lin, A recurrent neural network based health indicator for remaining useful life prediction of bearings, Neurocomputing 240 (2017) 98–109.

[32] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533.

[33] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[34] M. Henaff, A. Szlam, Y. LeCun, Recurrent orthogonal networks and long-memory tasks, 2016, arXiv preprint, arXiv:1602.06662.

[35] K. Cho, B.Van. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.

[36] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.

[37] S. Jean, K. Cho, R. Memisevic, Y. Bengio, On using very large target vocabulary for neural machine translation, 2014, arXiv preprint arXiv:1412.2007.

[38] M.T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, 2015, arXiv preprint arXiv:1508.04025.

[39] R. Jozefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, in: International Conference on Machine Learning, 2015, June pp. 2342-2350.

[40] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[41] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, C. Varnier, PRONOSTIA: An experimental platform for bearings accelerated degradation tests, in: IEEE International Conference on Prognostics and Health Management, PHM'12. IEEE Catalog Number: CPF12PHM-CDR. 2012, June, pp. 1-8.

[42] E. Sutrisno, H. Oh, A.S.S. Vasan, M. Pecht, Estimation of remaining useful life of ball bearings using data driven methodologies, in: Prognostics and Health Management, PHM, 2012 IEEE Conference on, IEEE, 2012, pp. 1–7.

[43] A.Z. Hinchi, M. Tkiouat, Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network, Procedia Comput. Sci. 127 (2018) 123–132.

[44] Y. Lei, N. Li, S. Gontarz, J. Lin, S. Radkowski, J. Dybala, A model-based method for remaining useful life prediction of machinery, IEEE Trans. Reliab. 65 (3) (2016) 1314–1326.

[45] S. Hong, Z. Zhou, E. Zio, K. Hong, Condition assessment for the performance degradation of bearing based on a combinatorial feature extraction method, Digit. Signal Process. 27 (2014) 159–166.