Measurement xxx (xxxx) xxx

Contents lists available at ScienceDirect

Measurement

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

An enhanced encoder–decoder framework for bearing remaining useful life prediction☆

Lu Liu [a], Xiao Song [b], Kai Chen [a,*], Baocun Hou [c], Xudong Chai [c], Huansheng Ning [d]

[a] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[b] School of Cyber Science and Technology, Beihang University, Beijing 100191, China
[c] Beijing Aerospace Smart Manufacturing Technology Development Limited Corporation, Beijing, China
[d] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

ARTICLE INFO

Keywords:
Encoder–decoder
Trigonometric functions
Cumulative operation
Fitness analysis

ABSTRACT

In recent years, data-driven approaches for remaining useful life (RUL) prognostics have aroused widespread concern. Bearings act as the fundamental component of machinery and their conditioning status is closely associated with the normal operation of equipment. Hence, it is crucial to accurately predict the remaining useful life of bearings. This paper explores the degradation process of bearings and proposes an enhanced encoder–decoder framework. The framework attempts to construct a decoder with the ability to look back and selectively mine underlying information in the encoder. Additionally, trigonometric functions and cumulative operation are employed to enhance the quality of health indicators. To verify the effectiveness of the proposed method, vibration data from PRONOSTIA platform are utilized for RUL prognostics. Compared with several state-of-the-art methods, the experimental results demonstrate the superiority and feasibility of the proposed method.

1. Introduction

As the fundamental component in the equipment manufacturing industry, bearings play a significant role in the Industrial Internet of Things (IIoT) domain. The health conditions of bearings directly determine the performance and reliability of major equipment and then raise uncertainty for IIoT. Once the bearings in critical positions have problems such as wear and crack, they will affect the operation of the entire mechanical system, making it unable to work properly, and even causing major accidents and casualties. Traditionally, there are two types of bearing maintenance strategies that involve post-failure maintenance and scheduled maintenance. It is quite clear that these two maintenance strategies will result in deficient maintenance or excessive maintenance. In recent years, condition-based maintenance (CBM) has aroused considerable attention due to its inherent ability to monitor the health conditions of bearings in real-time and make an optimal maintenance decision based on condition monitoring information [1]. Health prognostics is considered as one of the primary tasks in CBM, which aims at predicting the remaining useful life (RUL) of machinery based on the historical and on-going degradation patterns observed from condition monitoring information. The accurate RUL of the bearings helps to find the bearings' damage and defects as early as possible to carry out maintenance and repair, reduce equipment downtime, and enhance the reliability of machinery. Generally, most approaches developed for bearing RUL prognostics can be roughly grouped into three types: physical model-based methods [2,3], statistical model-based methods, and data-driven methods. Physical model-based methods attempt to build mathematical models based on failure mechanisms or the first principle of damage [4]. The Paris–Erdogan (PE) model [5] and its variants [6–8] are widely used physics models. The established model takes into account the impact of various factors (e.g., material, load force, lubrication) on the bearings, and the parameters of the model are modified through a significant amount of experimental data. The physical model-based approaches can provide accurate prognostics of RUL based on an intensive comprehension of the failure mechanisms and valid assessment of model parameters. However, in most cases, it is difficult to develop accurate mathematical models for a specific bearing, and the auxiliary cost is high as well. Statistical model-based approaches utilize empirical knowledge to predict the RUL of bearings [9]. Among the statistical approaches, RUL estimation models are developed by fitting available condition monitoring data into random coefficient models or process models based on a probabilistic method. There have been a variety of statistical models, including autoregressive (AR) models [10,11], random coefficient models [12,13], Wiener

☆ This work was supported by the National Key Research and Development Program of China under Grant 2018YFB1702703.
* Corresponding author.
E-mail address: chenkaivisual@buaa.edu.cn (K. Chen).

https://doi.org/10.1016/j.measurement.2020.108753
Received 12 August 2020; Received in revised form 8 November 2020; Accepted 15 November 2020
Available online 20 November 2020

process models [14,15], etc. These statistical model-based prediction methods highly rely on essential data of bearings. Moreover, when operating conditions of bearings make a difference, the estimation result of this approach tends to be less accurate for its inherent poor adaptability.

Recently, data-driven methods, which mainly employ artificial intelligence (AI) techniques to recognize the bearing degradation pattern, are becoming increasingly prevalent across various industries [16–18]. AI-based methods are free from constructing mathematical models or statistical models and just focus on the condition monitoring data. Many researchers have proposed all kinds of data-driven approaches concerning machinery RUL prognostics. Liu et al. [19] developed a support vector machine (SVM) to predict the machinery condition. To better cope with the irregular original signal, wavelet transform (WT) was introduced into the SVM model. Khelif et al. [20] proposed a direct remaining useful life estimation method using a support vector regression (SVR) model where the RUL can be estimated at any time instant of the degradation process. With the vast amounts of data and increasingly high dimension in data, traditional machine learning methods have a low capacity of mapping implicit relations between different features acquired from bearing raw data. Deep learning (DL) is becoming a new solution to the problems of complex data and deep representation. Convolutional neural network (CNN) and RNN (recurrent neural network) are commonly known DL models and have achieved much success in the field of computer vision, speech recognition, natural language processing, and machine translation. Motivated by the great ability of CNN, researchers have conducted related work about bearing diagnosis and bearing RUL prediction. Zhang et al. [21] presented a deep convolutional neural network to enhance the performance of bearing fault diagnosis because of the different working loads and inevitably noisy environments. Chen et al. [22] proposed a novel bearing fault diagnosis scheme that combined cyclic spectral coherence (CSCoh) and convolutional neural network to improve classification performance. Zhu et al. [23] introduced a new deep feature learning approach for bearing RUL prognostics through time-frequency representation (TFR) and multiscale convolutional neural network (MSCNN). Wang et al. [24] directly regarded the raw multi-sensor data as input for a deep separable convolutional network (DSCN) and effectively modeled the interrelationships of various sensors data. Compared with CNN, RNN is more suitable for time-series processing and capable of seeking underlying information from historical data. RNNs mentioned here involve traditional RNN and its variants such as long short-term memory (LSTM) and gated recurrent unit (GRU). Until now, many scholars have made some related work on bearing RUL prognostics using RNN models. Chen et al. [25] applied kernel principal component analysis (KPCA) to reduce the dimensionality of source data and extract nonlinear features and then presented a GRU to predict RUL of turbofan engines. Guo et al. [26] selected the most sensitive features from candidate features based on monotonicity and correlation metrics and then fed them into a recurrent neural network for RUL prediction of bearings. Ren et al. [27] combined the feature extraction methods and RUL model by pre-training the Restricted Boltzmann Machine (RBM) and proposed an end-to-end network namely Multi-scale Dense Gated Recurrent Unit Network (MDGRU).

Through the above retrospective reviews of data-driven approaches, RUL prediction of bearings faces two major challenges, i.e., health indicator (HI) construction, and prognostics model selection. Health indicator, commonly known as a feature that was extracted from the raw data, has a pivotal impact on the prognostic model. Most importantly, HIs that properly reflect failure progression can simplify the prognostic modeling and generate precise prognostic results. Generally, the data-driven prognostics need to undergo imperative steps of training and testing. Firstly, raw data are collected from sensors and then preprocessed to construct HIs to learn degradation behavior. Secondly, the trained model is used to test the prediction performance. According to literature, although most data-driven methods could handle the nonlinearity of degradation signals, features with monotonicity tend to bring about better RUL prognostics results. Besides, some original features may fail to reflect the degradation tendency. Hence, it is quite essential to construct proper features as input for the prediction model. Additionally, the accuracy of existing RNN-based prognostics models decreases when time-series data covers the whole life of the machine, as too long sequence will result in vanishing or exploding gradient problems. On the other hand, short time-series cannot reflect the internal relationship and evolution trend.

In order to overcome the aforementioned drawbacks, we propose an encoder–decoder structure with an attention mechanism for RUL prediction, which relies on cumulative features and the LSTM model. Given the fact that bearing degradation is a cumulative process, we are inspired to process features in a cumulative manner, thereby accomplishing the purpose of mining the overall data and obtaining transformed features with essential characteristics such as monotonicity and trendability. Besides, we employ trigonometric functions to improve the quality of the extracted features. Base on the LSTM structure, this paper proposes to extend the basic encoder–decoder architecture by allowing the decoder to automatically search for parts of a source sequence that are used to predict a sequence of RUL values. Afterward, by applying linear regression operation based on the least square method, the accurate remaining useful life of bearings can be obtained. The main contributions of this paper are summarized as follows.

(1) We employ trigonometric functions and use a cumulative transformation on several features in order to improve the characteristics of monotonicity and trendability, thereby providing more feasible inputs for the RUL prediction model.
(2) An LSTM-based encoder–decoder architecture with attention mechanism is presented to selectively mine historical information from original sequence data, which could realize the goal of estimating RUL automatically.
(3) The proposed RUL prediction method has been generalized on PRONOSTIA platform and the experimental results assert the superiority of the proposed RUL prognostics model.

The rest of this paper is organized as follows. Section 2 introduces the theoretical framework of the proposed method which mainly involves signal denoising, the scheme of cumulative transformation, and the encoder–decoder structure. Section 3 demonstrates the effectiveness of the presented RUL prediction method on a run-to-failure dataset. Finally, the conclusions are drawn in Section 4.

2. Theoretical framework

Generally, the data-driven methods consist of three key steps: signal denoising, feature extraction, and RUL prediction. Firstly, in order to eliminate harmful noisy parts, raw data acquired from sensors should be denoised through appropriate signal preprocessing. Then, in the stage of feature extraction, trigonometric functions are intended for extracting features and are then smoothed to remove unwanted noise. Meanwhile, the extracted features together with classical features are further transformed into their respective cumulative form. Finally, time-series data containing cumulative features are input into the LSTM-based encoder–decoder network which outputs the health status of bearing. By applying a linear regression algorithm on the outputs, the RUL value at the end of the collected signal can be calculated. The flowchart of the proposed method is illustrated in Fig. 1. Detailed techniques are described as follows.

2.1. Signal denoising

The commonly used condition monitoring (CM) data include vibration, acoustic, and temperature signals. Among these signals, the vibration signal is widely analyzed as it provides abundant information about the anomalies inside the bearings. The extraction of HI in such

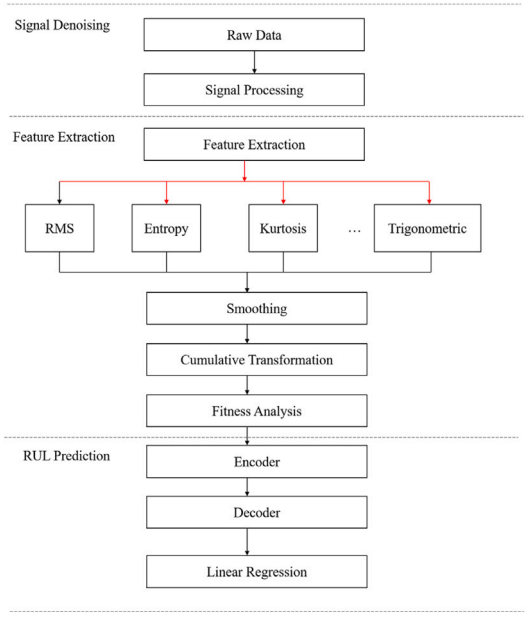


Fig. 1. Flowchart of proposed RUL prediction method.

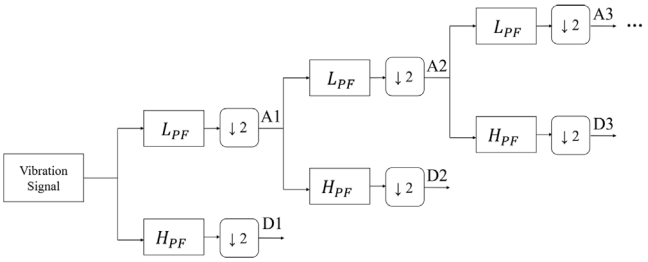


Fig. 2. The schematic map of wavelet decomposition.

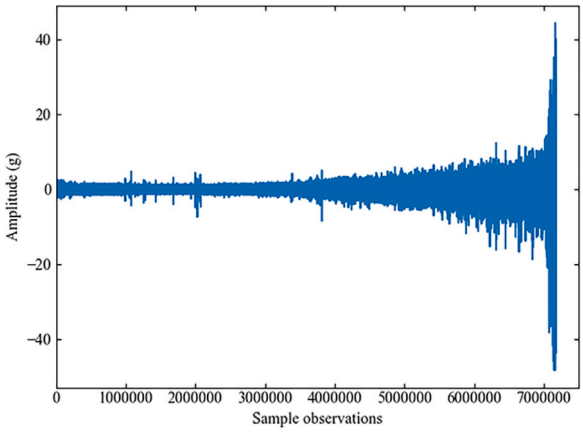


Fig. 3. The whole life of vibration signal from bearing.

signals can be done using the time domain, the frequency domain, and the time-frequency domain analyses. In practice, the time domain and the frequency domain analyses have been rarely used because of their inability to deal with nonstationary signals. time-frequency techniques tend to explore signals in both time and frequency domains. Wavelet transform (WT) is such an outstanding technology that is widely used in vibration content characterization. There exist some other popular time-frequency techniques including short-time Fourier transform (STFT) and empirical mode decomposition (EMD). STFT is unable to reveal inherent information of non-stationary signals which are common in bearing vibration signals. The whole life of vibration signal from some bearing is shown as Fig. 3. Also, EMD is often used in demodulation applications and has similar characteristics as WT. Herein, we adopt a discrete wavelet transform (DWT) technique to cope with vibration data. The most significant implementation of DWT is known as multiresolution analysis (MRA), which is fulfilled by two sets of functions, i.e., scaling function and wavelet function, that are associated with a low-pass filter L_{PF} and a high-pass filter H_{PF}, respectively. At the stage of the first level of decomposition, the original signal is firstly passed through an L_{PF} that gives an approximation coefficient $A1$ and through an H_{PF} that gives a detail coefficient $D1$, followed by down-sampling by 2. The same transformation on the approximation coefficient $A1$ makes it to be decomposed further into approximation $A2$ and detail $D2$. The approximation parts contain the high-scale, low-frequency components and the details contain the low-scale, high-frequency components of the signal. As illustrated in Fig. 2, the wavelet decomposition for level 3 can be formulated as

$$s = A3 + D3 + D2 + D1 \tag{1}$$

where s represents vibration signal, $A3$ is the approximation for level 3, and $D3$, $D2$, $D1$ are details for level 3, 2, 1, respectively. It is worth noting that MRA has three essential requirements, namely the mother wavelet, the decomposition level, and the features to be extracted. The main task of signal denoising is to process the detailed coefficients based on wavelet threshold functions. Herein, we employ a soft thresholding function and fixed threshold to deal with the detailed coefficients. The formulations of soft thresholding function and threshold are shown below.

$$w_\lambda = \begin{cases} sgn(w)(|w| - \lambda) & |w| \geq \lambda \\ 0 & |w| < \lambda \end{cases} \tag{2}$$

$$\lambda = \sqrt{2log(N)} \tag{3}$$

w_λ is the processed detailed coefficient, λ is the threshold and N is the length of detailed coefficients sequence. The principle of wavelet-based denoising is different from that of the conventional filter-based methods. In the filter-based methods, the frequency components outside a certain range are usually set to zero, which may drop some useful fault information since some fault signals serve as impulses and take up a wide frequency range. The filter-based methods will smooth some impulses. However, the wavelet-based denoising methods are often used to set small wavelet transform coefficient to zero, which preserve the impulses well because these impulses are viewed as big wavelet coefficients. Thereby, the wavelet-based methods are more appropriate for preprocessing of fault signals than filter-based methods.

2.2. Feature extraction and smoothing

The classical approach for feature extraction with DWT is developed at a required level of decomposition to extract various features. The specific procedure is divided into three steps: wavelet decomposition, threshold processing on detailed coefficients, and wavelet reconstruction. In literature, various existing deep prognostics networks still require manually extract and select some discriminative features from the reconstructed signals. The extracted features can be RMS, kurtosis, skewness, or other statistical features. Nevertheless, each feature can be sensitive to different damage. For example, RMS and kurtosis from the degrading bearings demonstrate obvious variation only a few times before failure, which restricts the usage of such features for RUL prediction. In this case, a set of features are presented as useful supplements, e.g., inverse hyperbolic sine (asinh) and inverse tangent (atan). These two trigonometric functions are considered as the most promising options because their domain $(-\infty, +\infty)$ fits the real vibration data well and they can transform the raw data into lower scales.

Table 1
Features and corresponding formulas.

Feature	Formula
SD of atan(X)	$X_{atan} = \sigma\left(log[x_j + \sqrt{x_j{}^2 + 1}]\right)$
SD of asinh(X)	$X_{asinh} = \sigma\left(\frac{i}{2} log\left(\frac{i + x_j}{i - x_j}\right)\right)$
Standard deviation (SD)	$X_{sd} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \mu\right)^2}$
Peak	$X_p = max\|X\|$
Peak-to-peak	$X_{p-p} = max\|X\| - min\|X\|$
Root mean square (RMS)	$X_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i{}^2}$
Upper bound	$X_{upper} = max\|X\| + \frac{1}{2}\frac{max\|X\| - min\|X\|}{n-1}$
Impulse factor	$X_{if} = \frac{X_p}{\frac{1}{n}\sum_{i=1}^{n}\|x_i\|}$
Crest factor	$X_{cf} = \frac{X_p}{X_{rms}}$
Margin factor	$X_{mf} = \frac{X_p}{(\frac{1}{n}\sum_{i=1}^{n}\sqrt{\|x_i\|})^2}$
Energy	$X_e = \sum_{i=1}^{n} x_i{}^2$
Kurtosis	$X_k = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \mu\right)^4}{X_{sd}{}^4}$
Skewness	$X_{sk} = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \mu\right)^3}{X_{sd}{}^3}$

The transformation is performed as follows. A trigonometric function operates on an array X elementwise (x_j, $j = 1, 2, \ldots,$ n), and the standard deviation (SD) is carried out to scaled array for extraction of the feature. The features extracted from the denoised vibration data are listed in Table 1. Although these features are developed based on expert knowledge and proven to be effective in some applications, cumulative transformations are still designed to enhance the monotonicity of HIs. Prior to the accumulation of measurements, a smoothing process is indispensable to reduce the fluctuation of the extracted features and further filter unwanted noise. Herein, we apply Savitzky–Golay filter to increase the precision of the data without distorting the signal tendency. The smoothing process is achieved by fitting successive subsets of adjacent data points with a low-degree polynomial using the method of linear least squares.

2.3. Feature cumulative transformation

It is well known that most machinery will inevitably undergo the degradation process. Features that can reflect the degradation trends of bearings contribute to accurate prognostics. The existence of degradation process motivates us to select a feature transformation method since the original features are often unsatisfactory and fail to reflect the degradation trend. An effective method to construct HI is presented by transforming an extracted feature into its accordingly cumulative form. Specifically, the feature transformation is conducted by applying a cumulative function on a time-series where the pointwise running total and scaling operations are executed at the same time with cumulative features following which can characterize the trend of degradation. The cumulative function is performed as follows.

$$cf_{jn} = \frac{\sum_{i=1}^{n} f_j(i)}{\sqrt{\left|\sum_{i=1}^{n} f_j(i)\right|}}, j = 1, 2, \ldots, N \tag{4}$$

cf_{jn} represents a running total of jth feature $f_j(i)$ up to n samples of feature. Most notably, the proposed cumulative algorithm can be easily affected by unnecessary noise; thus, it is essential to smooth the features before transformation. After the cumulative transformation, it is crucial to design metrics to evaluate the quality of features. In general, there are two common metrics evaluating suitability of the constructed features for RUL prognostics, i.e., monotonicity and trendability.

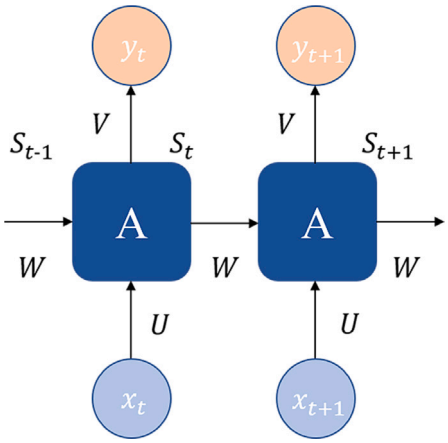


Fig. 4. The structure of RNN.

Monotonicity reflects the implicit increasing or decreasing trend of features, and it is viewed as a crucial element of a degradation feature. It can be calculated via a single HI as follows:

$$M = \frac{\left|No.\,of\,\frac{d}{dx} > 0 - No.\,of\,\frac{d}{dx} < 0\right|}{n-1} \tag{5}$$

n represents total observations. $M = 1$ and $M = 0$ stands for highly and badly monotonic trend of features, respectively.

Trendability is measured to describe how the degradation status changes with runtime. In other words, this metric characterizes a function which is concerned with feature and its correlation with time. Its formulation is drawn as follows:

$$T = \frac{\left|\sum_{i=1}^{N}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)\right|}{\sqrt{\sum_{i=1}^{N}\left(x_i - \overline{x}\right)^2\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2}} \tag{6}$$

where x_i and y_i represent the ranks of the time and the feature, respectively. N is the length of time sequence. The correlation coefficient T ranges from 0 to 1 and higher value implies a strong monotonic relationship between health indicator and time.

2.4. Prognostics model construction

The architecture of a simple RNN is illustrated in Fig. 4. It can be seen that connections between nodes form a directed graph along a temporal sequence. In other words, RNN neuron receives two values, i.e., input at this moment and hidden state from the previous moment, then outputs y_t and updated hidden state S_t. The formulations that describe the process is listed as follows:

$$S_t = g\left(U \cdot x_t + W \cdot S_{t-1} + b_g\right) \tag{7}$$

$$y_t = f\left(V \cdot b_f\right) \tag{8}$$

where $g(\cdot)$ and $f(\cdot)$ are activation functions, and U, V, W are the matrices of weights among the input layer, output layer, and hidden layer. The vectors b_g and b_f are bias parameters that enable each node to learn an offset. Compared with feedforward networks, RNNs play as an extension to process variable-length sequences. However, with the recursive progress, RNNs are often confronted with the problems of vanishing or exploding gradients and fail to capture the correlations between various inputs at different time steps.

In fact, RNNs are a class of neural networks that are naturally suitable for processing time-series data. As a variant of traditional RNN, LSTM has a similar chain-like structure, but the repeating module (also called a cell) has a complex structure. The appearance of LSTM provides a new solution to a large variety of problems where traditional RNN

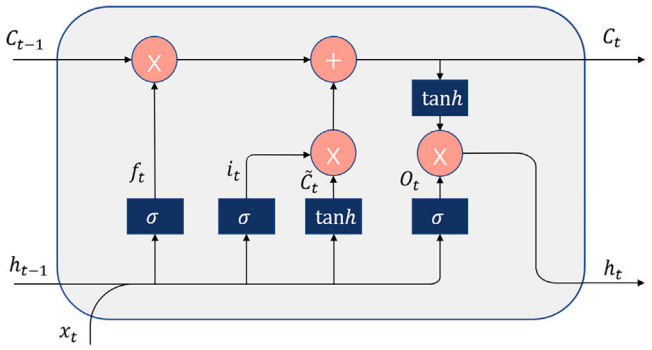


Fig. 5. The structure of LSTM.

cannot work well. The structure of LSTM is illustrated in Fig. 5. LSTM has the superiority of handling long-term sequence data for the reason that it can remove or add information to the cell state by unique structures called gates. The gates provide an approach for optionally filtering information. As shown in Fig. 5, the sigmoid function $\sigma(\cdot)$ and pointwise multiplication operation $\otimes$ act as the gates. The detailed principle of LSTM is introduced below. Firstly, LSTM receives the value of h_{t-1} and x_t, following by a sigmoid layer called "forget gate" which outputs a number between 0 and 1. The output value then operates on the cell state C_{t-1} and decides how much information should be left. The number 0 stands for "totally keep this" while number 1 represents "get rid of this". Secondly, LSTM needs to decide what information should be stored in the cell state. A tanh layer generates a vector of candidate values C_t and then a sigmoid layer called "input gate" is used to determine which values of C_t to be selected. Next, it is time to update the precious cell state C_{t-1} into a new cell state C_t. The update operation includes two steps: multiplying the old cell state C_{t-1} by f_t, and adding the new candidate values $i_t \otimes C_t$. Finally, we need to decide the output based on the new cell state C_t. Herein, a sigmoid layer called "output gate" is constructed to decide what parts of the cell state should be output. In Fig. 5, it can be found that the cell state C_t passes through a tanh layer and multiply it by the output of the "output gate". The whole mathematical expressions are formulated below.

$$f_t = \sigma\left(W_{fh}h_{t-1} + W_{fx}x_t + b_f\right) \tag{9}$$

$$i_t = \sigma\left(W_{ih}h_{t-1} + W_{ix}x_t + b_i\right) \tag{10}$$

$$\tilde{C}_t = tanh\left(W_{Ch}h_{t-1} + W_{Cx}x_t + b_C\right) \tag{11}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{12}$$

$$O_t = \sigma\left(W_{Oh}h_{t-1} + W_{Ox}x_t + b_O\right) \tag{13}$$

$$h_t = O_t \otimes tanh\left(C_t\right) \tag{14}$$

$W_{fh}, W_{fx}, W_{ih}, W_{ix}, W_{Ch}, W_{Cx}, W_{Oh}, W_{Ox}$ are matrices of weights among different layers. b_f, b_i, b_C, b_O are bias parameters.

2.5. Encoder–decoder architecture

RNN encoder–decoder is firstly proposed to learn phrase representation for statistical machine translation. A classical encoder–decoder structure is illustrated in Fig. 6. The proposed neural network architecture is composed of two recurrent neural networks (RNNs) which are referred to as an encoder and a decoder pair. The encoder is designed to transform a variable-length source sequence into a fixed-dimensional vector representation c, and then the decoder extracts another variable-length sequence from the vector representation. The conditional probability of y_t is modeled as follows:

$$p\left(y_t|y_{t-1}, y_{t-2}, \ldots, y_1, c\right) = g\left(y_{t-1}, h_t, c\right) \tag{15}$$

$$h_t = f\left(y_{t-1}, h_{t-1}, c\right) \tag{16}$$

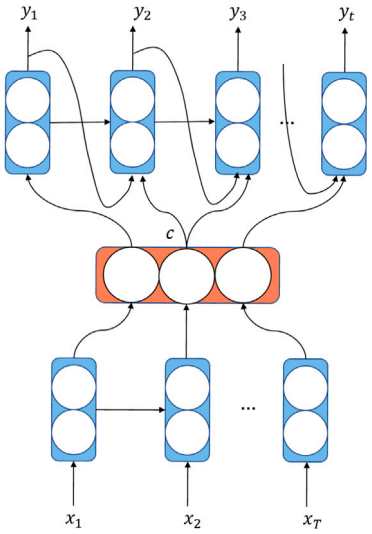


Fig. 6. The encoder–decoder architecture.

where $f(\cdot)$ and $g(\cdot)$ are activation functions. In the application of encoder–decoder, the length of the source sequence can be different from that of the target sequence, resulting in a huge boost in flexibility.

Nevertheless, RNN encoder–decoder has trouble coping with long-range temporal dependencies due to the vanishing or exploding gradients of RNN. Besides, the decoder mentioned above receives the same vector representation c at each time step, which means that the decoder merely obtains the global information from the encoder rather than optionally extracts information from the output of the encoder. Many scholars pay attention to the problem and make attempts to employ the output of encoder with an attention mechanism. The architecture of LSTM-based encoder–decoder with attention mechanism is illustrated in Fig. 7.

Unlike the existing encoder–decoder method, we redefine each conditional probability as:

$$p\left(y_t|y_{t-1}, y_{t-2}, \ldots, y_1, c\right) = g\left(y_{t-1}, h_t, c_j\right) \tag{17}$$

$$h_t = f\left(y_{t-1}, h_{t-1}, c_j\right) \tag{18}$$

where the probability is based on a unique vector representation c_j for every target y_j. The vector representation c_j is computed as a weighted sum of output of LSTM neural networks which are denoted as h_{x1}, h_{x2} and so on.

$$c_j = \sum_{i=1}^{T} w_{ji}h_{xi} \tag{19}$$

The weight w_{ji} is calculated by

$$w_{ji} = \frac{exp\left(e_{ji}\right)}{\sum_{k=1}^{T} exp\left(e_{jk}\right)} \tag{20}$$

$$e_{ji} = score\left(h_{xi}, h_{yj}\right) \tag{21}$$

where e_{ji} is an alignment model which scores the correlation between hidden states h_{xi} and h_{yj} from encoder and decoder, respectively. Herein, in order to mine underlying information, we concatenate c_j and h_{yj} together and input them into the following fully-connected neural network.

3. Experiment verification

3.1. Dataset description

In order to validate the effectiveness and superiority of the proposed RUL prognostics approach, we employ bearing datasets from an experimental platform called PRONOSTIA (Fig. 8), which were provided by

Table 2
Datasets of IEEE 2012 PHM prognostic challenge.

Operation conditions	Radial force (N)	Rotating speed (rpm)	Learning set	Test set		
Condition 1	4000	1800	Bearing 1_1 Bearing 1_2	Bearing 1_3 Bearing 1_6	Bearing 1_4 Bearing 1_7	Bearing 1_5
Condition 2	4200	1650	Bearing 2_1 Bearing 2_2	Bearing 2_3 Bearing 2_6	Bearing 2_4 Bearing 2_7	Bearing 2_5
Condition 3	5000	1500	Bearing 3_1 Bearing 3_2	Bearing 3_3		

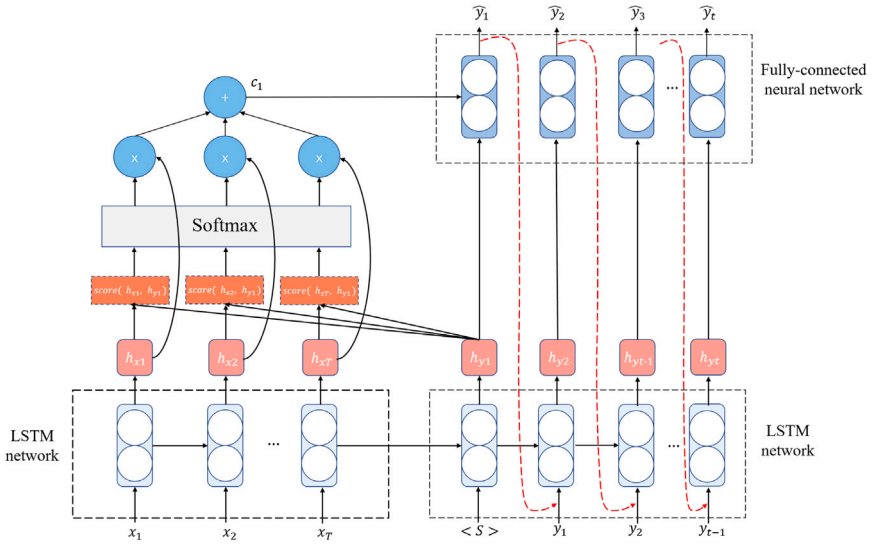


Fig. 7. The encoder–decoder architecture with an attention mechanism.

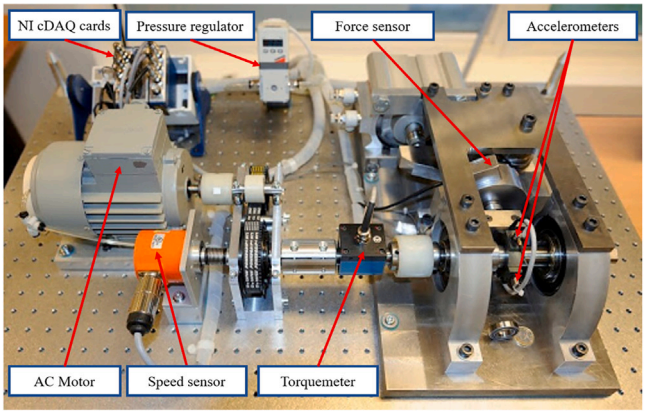


Fig. 8. The PRONOSTIA platform.

FEMTO-ST Institute and utilized in IEEE PHM 2012 Prognostic Challenge. The PRONOSTIA enables accelerated degradation of bearings under three different operating conditions, and a total of seventeen run-to-failure datasets including six training datasets and eleven testing datasets are offered. The detailed information about the datasets is listed in Table 2. In the experiments, the bearing's condition monitoring is ensured by collecting two types of signals: temperature and vibration (horizontal and vertical accelerometers). In literature, the horizontal vibration signals commonly provide more useful information than the vertical ones for revealing the trend of bearing degradation. As a result, we extract features from the horizontal vibration signals. The sampling frequency of the vibration signal is 25.6 kHz, and 2560 samples, i.e., 1/10 s, are recorded every 10 s.

3.2. Metrics for RUL prognostics

To compare the presented RUL prognostics method with state-of-the-art method, it is necessary to build up metrics. In the IEEE PHM 2012 Challenge, a score function is constructed as below:

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

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

$$score = \frac{1}{11}\sum_{i=1}^{11} A_i \tag{24}$$

where $actRUL_i$ denotes actual remaining useful life of ith bearing, and $\widehat{RUL_i}$ represents the prediction result of bearing RUL. The relationship between percentage error Er_i and score A_i is drawn as follows in Fig. 9.

3.3. Feature extraction and fitness analysis

According to previous descriptions in Section 2, the decomposition of the vibration signal requires the mother wavelet and a decomposition level. There are various wavelet functions available such as Haar, Daubechies, Meyer, and Morlet wavelets. They have different characteristics and are used for different purposes. The Haar wavelet is naturally not continuously differentiable which limits its applications. The Meyer wavelet is continuously differentiable, whereas it does not have compact support. Morlet wavelet is a type of nonorthogonal wavelet and it does not have compact support as well. In general, Daubechies wavelets are the most appropriate functions for multiresolution analysis because they have the properties of orthogonality and compact support. Daubechies wavelets are mainly used in discrete wavelet transform (DWT) and usually used in digital signal analysis, signal compression, and noise removal. In this paper, we apply a

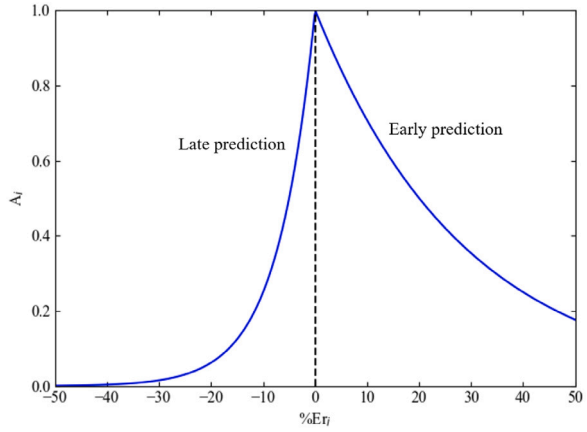


Fig. 9. The scoring function of a RUL estimates according to its percent error.

Daubechies wavelet of the fourth-order (db4) as some literature used to do. After the step of signal denoising, it is supposed to extract statistical features, apply trigonometric functions, and perform the cumulative transformation. In Fig. 10(a) and (c), the classical features show low monotonicity/trendability and high scales. Hence, it is necessary to consider the scheme of feature extraction utilizing an integration of standard deviation and trigonometric functions (see Table 1). The results in Fig. 10(b) and (d) show that the trigonometric features effectively reveal high monotonicity and trendability with the gradual evolution of degradation. Additionally, the trigonometric features clearly have lower scales in comparison with the classical features, which are beneficial for the training process of the prognostics model. Although some classical features cannot exhibit a clear trend of bearing degradation, it is proven to be effective by operating a cumulative transformation on the nonideal features. From Fig. 11(a) and (c), it is qualitatively observed that the forms of features without cumulative transformation appear to be low monotonicity and the trends differ from one other. On the contrary, after applying the cumulative transformation, the forms of features turn to be more monotonic and trendable by considering Fig. 11(b) and (d). It is obvious that the cumulative transformation has a strong ability to correct the trend of features. More specifically, Figs. 12 and 13 illustrate the histograms of each feature from training datasets that represent their monotonicity and trendability before and after cumulative transformation. The mean performance of the cumulative transformation is listed in Table 3. It can be seen that cumulative features have higher fitness compared with their classical form. As discussed earlier, higher values of monotonicity and trendability contribute to accurate estimation. In this way, we abandon the values of energy and skewness to reduce the amount of calculation. The visual filtered result is shown in Fig. 14.

3.4. RUL prognostics process

In this part, the training and testing processes are further discussed. In the beginning, it is indispensable for any machine learning algorithm to make clear the input and output of the network. In this experiment,

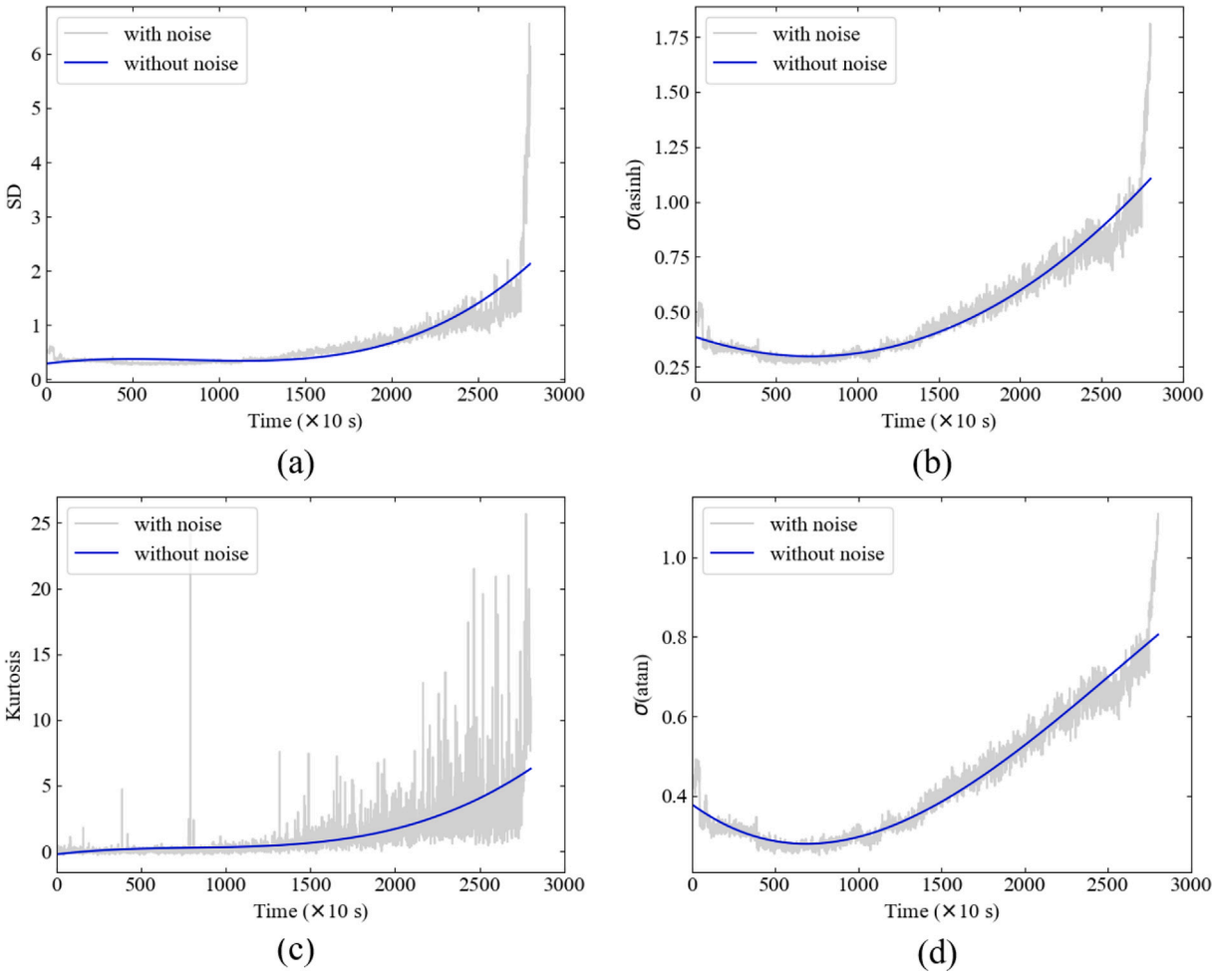


Fig. 10. Comparisons between classical features and trigonometric features in Bearing 1_1. (a) and (c) Classical features. (b) and (d) Trigonometric features.

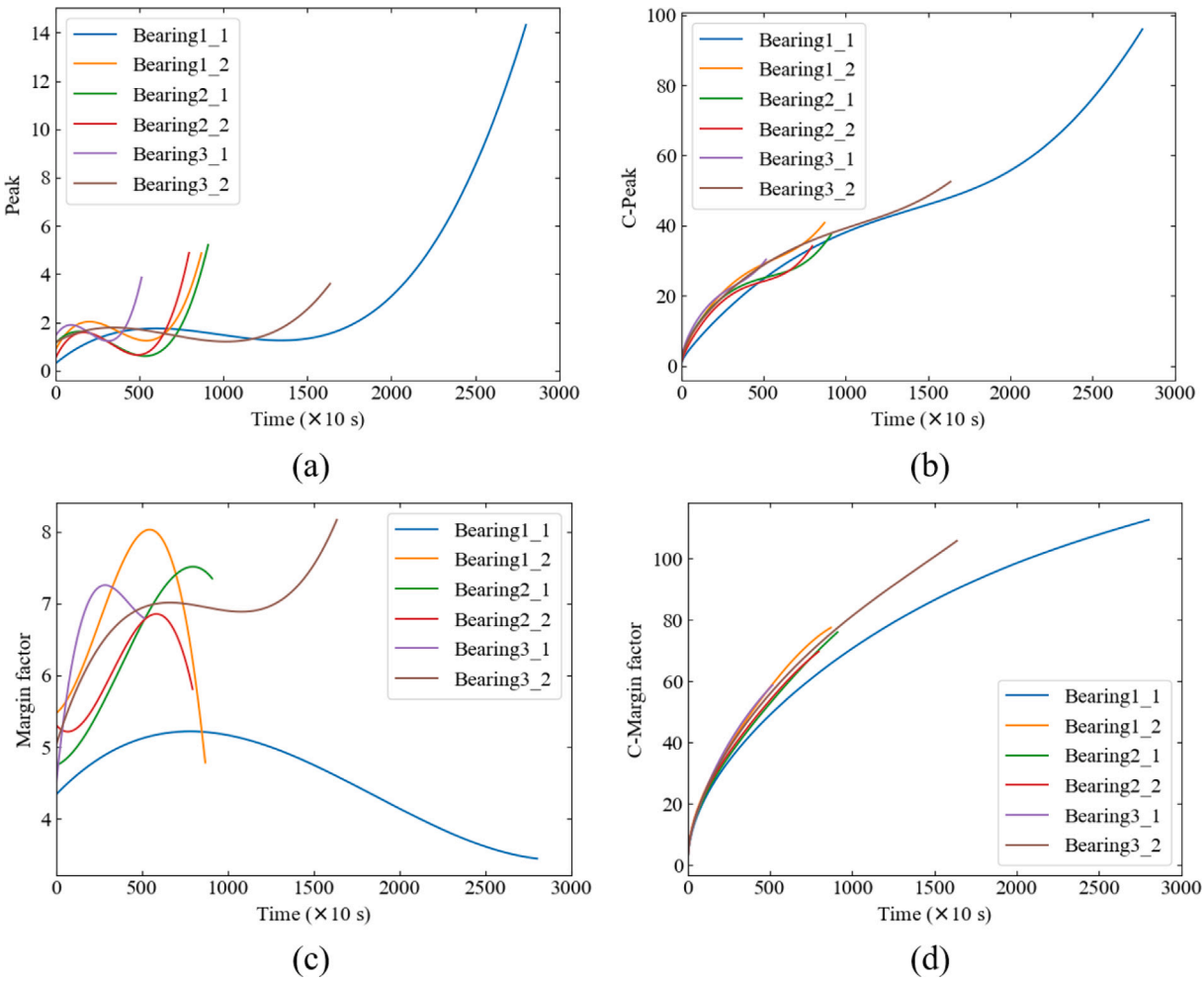


Fig. 11. The trend of classical features before and after cumulative transformation. (a) and (c) Classical features. (b) and (d) Cumulative features.

Table 3
Comparison of feature fitness.

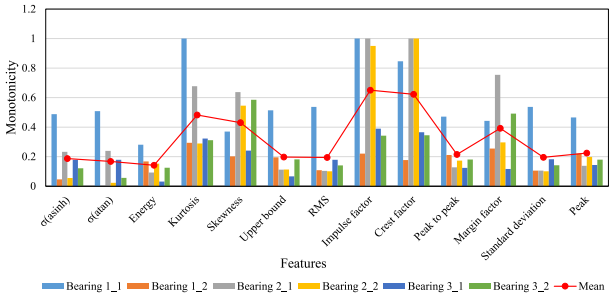
Feature	M	T	Cumulative feature	M	T
σ(atan)	0.17	0.34	C-σ(atan)	1	1
σ(asinh)	0.19	0.29	C-σ(asinh)	1	1
$\sigma(X)$	0.20	0.22	C-$\sigma(X)$	1	1
Peak	0.22	0.29	C-Peak	1	1
Peak-to-peak	0.21	0.26	C-Peak-to-peak	1	1
RMS	0.19	0.21	C-RMS	1	1
Upper bound	0.20	0.24	C-Upper bound	1	1
Impulse factor	0.65	0.79	C-Impulse factor	1	1
Crest factor	0.62	0.76	C-Crest factor	1	1
Margin factor	0.39	0.70	C-Margin factor	1	1
Energy	0.14	0.13	C-Energy	0.68	0.90
Kurtosis	0.48	0.66	C-Kurtosis	0.86	0.99
Skewness	0.43	0.70	C-Skewness	0.81	0.93

the inputs are a sequence of the vector that contains the cumulative features filtered in the last section, i.e., 11 features are thrown into the encoder at every time-step. The length of our model is designed as 10 because too short length makes it hard to take advantage of the attention mechanism. At the same time, too long sequence length will inevitably bring a vast quantity of computation to CPU/GPU, and reduce the operation speed. Therefore, the length of 10 is a proper choice. The output of the decoder is developed to indicate the degradation of bearings which range from 0 to 1. Herein, we construct a reasonable health index that is correlated to the RUL. As we know, RUL can be described as the length between the current inspection time and the breakdown time. Given the concept of RUL, a new simple but efficient health index can be calculated as follows.

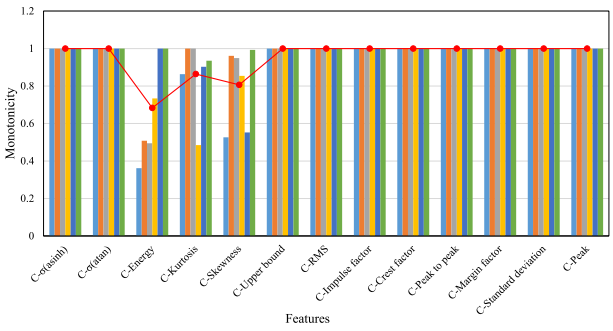
$$HI_t = \frac{actRUL_t}{actRUL_0} \tag{25}$$

$actRUL_0$ stands for the initial remaining useful life, namely the whole life. The health index ranges from 1 down to 0. In this case, the degradation of bearing is simplified as a linear process. For instance, each feature extracted from bearing 1_1 has 2803 sample points, then the $actRUL_0$ equals 2802 multiplied by the interval period. After one interval period, the RUL of bearing is 2801 multiplied by the interval period. In this way, the health index is calculated as 2801/2802. The specific training and testing processes are described below.

As the aforementioned section introduces, the encoder and decoder parts involve several LSTM cells, thus capable of dealing with temporal sequence problem. At the stage of training, the encoder looks at a sequence with a length of 10 step by step and the last LSTM cell generates a hidden state which contains the information of the whole sequence. In Fig. 7, it is worth noting that, during the training phase, the values of x_1, x_2 up to x_T, $\langle S\rangle, y_1, y_2$ up to y_{t-1} are input into the encoder and decoder respectively. After obtaining the hidden state that the decoder produces, a correlation between the hidden states from decoder and encoder is computed as a weight for measuring how closely they are. $\langle S\rangle$ is viewed as a start symbol of the decoder. $\langle S\rangle$ has the same shape with other inputs of the decoder. Herein, we make $\langle S\rangle$ just contain zero elements. During the test phase, the data flow among the decoder is illustrated as red dotted curves. The output from the last moment serves as the input of the following moment. In our method,

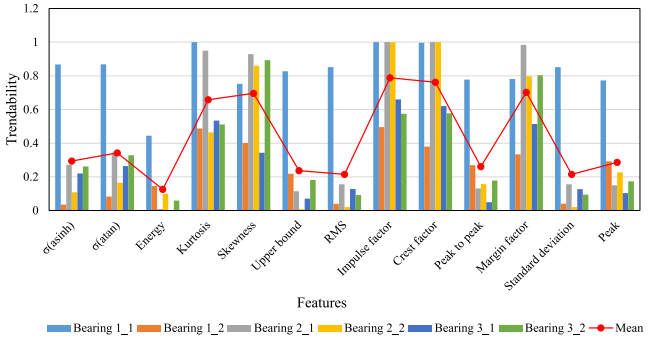


(a) Monotonicity of features before cumulative transformation.

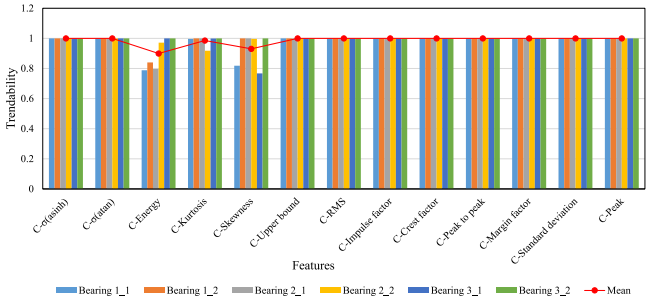


(b) Monotonicity of features after cumulative transformation.

Fig. 12. The monotonicity results of features from training dataset.



(a) Trendability of features before cumulative transformation.



(b) Trendability of features after cumulative transformation.

Fig. 13. The trendability results of features from training dataset.

a mean square error is used as the loss function. After the health index is predicted, there is a predicted RUL value at every time-step. By combining these samples and fitting them using linear regression, a straight line is drawn and the whole lifetime can be formulated when RUL drops down to 0.

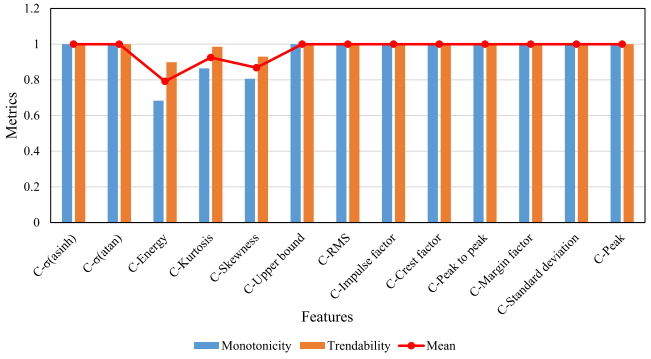


Fig. 14. Metrics for filtering features.

In addition, a fundamental assumption in the field of machine learning is that the testing data should follow the distribution of training data. The training data from PRONOSTIA are obtained by run-to-failure experiments, while the testing data only undergo the fragment of whole life. Therefore, to guarantee the common characteristics of training data and testing data, we use a random length of the original training data. Besides, a dropout is utilized to avoid overfitting. The training and testing processes are achieved in Python 3.6 with Pytorch 1.4.0, based on the platform with CPU as i7-9750H and GPU as GTX 1650. The learning rate of the training process is set as 0.001 and the epoch is 150.

To validate the superiority and effectiveness of the proposed method comprehensively, this paper introduces the mean absolute error value (MAE) and another metrics: score*100/MAE. This novel metrics meets our demand because a higher score and lower MAE certainly results in a higher ratio. Table 4 lists the specific values of RUL prediction using the proposed method and state-of-the-art methods. It is clear that the proposed method has achieved various degrees of improvement no matter in MAE, score, or score×100/MAE. Take an example for literature [28], the MAE value of our approach has dropped 31.41%. Meanwhile, the values of score and score×100/MAE are promoted by 4.23% and 51.95%, respectively. The promising result benefits from the advantage of attention mechanism and strong trend ability of cumulative features. The visual prediction results are illustrated in Fig. 15. It can be seen that the RUL prediction curves approximately fit the linear trend, which indicates that the proposed method has a better ability to learn the degradation. The encoder–decoder architecture with attention mechanism is proven to be an efficient strategy for bearing RUL prognostics. Cumulative features ensure the model to mine underlying relations between raw vibration and the health conditioning of bearings. It is satisfying that most of the bearings from the test set have learned the degradation trend, while some bearings have not. The prediction result of bearing 2_6 curves up at the end of the test. One of the reasons is that the data distribution of some of the test set is inconsistent with the learning set. Moreover, the quantity of learning set may be insufficient to exploit the underlying relationship between health indicators and RUL.

In retrospect, the accurate RUL prediction of the proposed method mainly benefits from three aspects, i.e., trigonometric functions, cumulative transformation, and an attention mechanism. To study their respective contributions, we have devised ablation experiments. Actually, the attention mechanism is designed to learn the correlation between current status and historical information. The principle of cumulative transformation is that current status involves all of the historical information. The combination of attention mechanism and cumulative transformation makes it possible to learn useful information selectively. In addition, trigonometric features have higher monotonicity

Table 4
RUL prognostics results and comparisons.

Testing datasets	Current time (s)	Actually RUL (s)	Predict RUL (s)	Ours	Er [28]	Er [29]	Er [30]	Er [26]	Er [31]
Bearing 1_3	18 010	5730	3650	36.30	7.62	37	54.73	43.28	−1.04
Bearing 1_4	11 380	2890	2500	13.49	69.77	80	38.48	61.94	85.81
Bearing 1_5	23 010	1610	1570	2.48	−72.57	9	−99.4	−22.98	−278.26
Bearing 1_6	23 010	1460	1640	−12.33	0.93	−5	−120.07	21.23	19.18
Bearing 1_7	15 010	7570	3210	57.60	85.99	−2	70.65	17.83	−7.13
Bearing 2_3	12 010	7530	2420	67.86	81.24	64	75.53	37.84	10.49
Bearing 2_4	6110	1390	1280	7.91	9.04	10	19.81	−19.42	51.8
Bearing 2_5	20 010	3090	4380	−41.74	28.19	−440	8.2	54.37	28.8
Bearing 2_6	5710	1290	1250	3.10	24.92	49	17.87	−13.95	−20.93
Bearing 2_7	1710	580	430	25.86	19.06	−317	1.69	−55.17	44.83
Bearing 3_3	3510	820	770	6.10	2.09	90	2.93	3.66	−3.66
$\overline{\lvert E_r \rvert}$				**24.98**	36.42	100.27	46.31	31.97	50.18
Score				**0.4654**	0.4465	0.3066	0.3831	0.2650	0.3547
Score × 100/$\overline{\lvert E_r \rvert}$				**1.8629**	1.2260	0.3058	0.8273	0.8289	0.7069

Table 5
RUL estimation results of ablation experiments.

Testing datasets	Trigonometric + cumulative + attention Error (%)	Without trigonometric functions Error (%)	Without cumulative transformation Error (%)	Without attention mechanism Error (%)
Bearing 1_3	36.30	27.44	−760.93	34.81
Bearing 1_4	13.49	−2.24	−454.053	16.77
Bearing 1_5	2.48	−16.50	11.65	−19.72
Bearing 1_6	−12.33	−33.34	−727.77	−37.72
Bearing 1_7	57.60	50.90	−6.36	56.83
Bearing 2_3	67.86	62.13	24.07	67.11
Bearing 2_4	7.91	−7.58	571.24	5.70
Bearing 2_5	−41.74	−64.15	−312.85	−43.89
Bearing 2_6	3.10	−11.80	−126.65	1.44
Bearing 2_7	25.86	18.10	951.46	24.89
Bearing 3_3	6.10	8.18	−181.74	3.84
$\overline{\lvert E_r \rvert}$	**24.98**	27.49	357.34	28.39
Score	**0.4654**	0.2653	0.1378	0.3853
Score × 100/$\overline{\lvert E_r \rvert}$	**1.8629**	0.9653	0.0367	1.3568

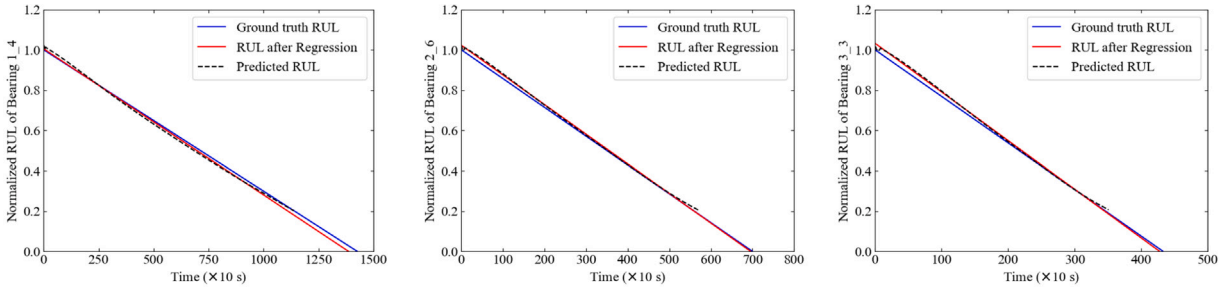


Fig. 15. The visualized prediction results of testing bearings.

and trendability than classical features, which is the ideal case for RUL prognostics. The application of trigonometric functions also reduces the scale of features which is beneficial for model training. Thereby, from Table 5, we can find that the proposed approach achieves the best performance among all the approaches.

4. Conclusion

With the widespread popularity of deep learning, a growing number of researchers seek for smarter solutions to conventional problems. RUL prognostics of bearing tends to be resolved in various data-driven methods instead of physics model-based or statistical model-based approaches. Encoder–decoder structure with attention mechanism is always be used in machine translation for its capability of dealing with long dependency problems. In this paper, we present a bearing RUL prediction model based on cumulative features and encoder–decoder structure. Firstly, trigonometric functions are operated on the denoised vibration data with two highly monotonic and trendy features produced. Combined with classical features, all features are transformed into respective cumulative forms. Then, to ensure the effectiveness of candidate features, a fitness analysis based on monotonicity and trendability is conducted to filter features below the threshold. Following that, the qualified features are feed into the encoder step by step. The attention mechanism is carried out by calculating the correlation between the hidden states of encoder and decoder. This key step allows the decoder to look back at the information of the encoder and decide what information to grasp. To verify the proposed method, its performance is compared with state-of-the-art contributions. The results clearly demonstrate the feasibility of the developed model. In future work, we will promote the generalization ability of the proposed model, because we noticed that performances on different bearings vary greatly even under the same operating conditions. The insufficiency of the model's generalized ability attributes to the small quantity of

training data for its inability to mine comprehensive relationships. Moreover, it is worthy of studying the interpretability of machine learning.

CRediT authorship contribution statement

Lu Liu: Conceptualization, Methodology, Software, Investigation, Writing - original draft. **Xiao Song:** Validation, Formal analysis, Visualization, Writing - review & editing, Software. **Kai Chen:** Validation, Formal analysis, Visualization, Writing - review & editing. **Baocun Hou:** Resources, Writing - review & editing, Supervision, Data curation. **Xudong Chai:** Resources, Writing - review & editing, Supervision, Data curation. **Huansheng Ning:** Writing - review & editing, Supervision, Data curation.

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.

References

[1] Z. Liu, L. Zhang, A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings, Measurement 149 (2020) 107002, http://dx.doi.org/10.1016/j.measurement.2019.107002.

[2] Y. Lu, Q. Li, S.Y. Liang, Physics-based intelligent prognosis for rolling bearing with fault feature extraction, Int. J. Adv. Manuf. Technol. 97 (1–4) (2018) 611–620, http://dx.doi.org/10.1007/s00170-018-1959-0.

[3] H. Hanachi, W. Yu, I.Y. Kim, J. Liu, C.K. Mechefske, Hybrid data-driven physics-based model fusion framework for tool wear prediction, Int. J. Adv. Manuf. Technol. 101 (9–12) (2019) 2861–2872, http://dx.doi.org/10.1007/s00170-018-3157-5.

[4] A.K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mech. Syst. Signal Process. 20 (7) (2006) 1483–1510, http://dx.doi.org/10.1016/j.ymssp.2005.09.012.

[5] P. Paris, F. Erdogan, A critical analysis of crack propagation laws, J. Fluids Eng. 85 (4) (1963) 528–533, http://dx.doi.org/10.1115/1.3656900.

[6] J. Wang, R.X. Gao, Z. Yuan, Z. Fan, L. Zhang, A joint particle filter and expectation maximization approach to machine condition prognosis, J. Intell. Manuf. 30 (2) (2019) 605–621, http://dx.doi.org/10.1007/s10845-016-1268-0.

[7] 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, http://dx.doi.org/10.1109/TR.2016.2570568.

[8] L. Liao, Discovering prognostic features using genetic programming in remaining useful life prediction, IEEE Trans. Ind. Electron. 61 (5) (2013) 2464–2472, http://dx.doi.org/10.1109/TIE.2013.2270212.

[9] 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, http://dx.doi.org/10.1016/j.ejor.2010.11.018.

[10] Y. Qian, R. Yan, S. Hu, Bearing degradation evaluation using recurrence quantification analysis and Kalman filter, IEEE Trans. Instrum. Meas. 63 (11) (2014) 2599–2610, http://dx.doi.org/10.1109/TIM.2014.2313034.

[11] C.K. Pang, J.-H. Zhou, H.-C. Yan, Pdf and breakdown time prediction for unobservable wear using enhanced particle filters in precognitive maintenance, IEEE Trans. Instrum. Meas. 64 (3) (2014) 649–659, http://dx.doi.org/10.1109/TIM.2014.2351312.

[12] X. Jin, Y. Sun, Z. Que, Y. Wang, T.W. Chow, Anomaly detection and fault prognosis for bearings, IEEE Trans. Instrum. Meas. 65 (9) (2016) 2046–2054, http://dx.doi.org/10.1109/TIM.2016.2570398.

[13] N.Z. Gebraeel, M.A. Lawley, R. Li, J.K. Ryan, Residual-life distributions from component degradation signals: A bayesian approach, IiE Trans. 37 (6) (2005) 543–557, http://dx.doi.org/10.1080/07408170590929018.

[14] K. Doksum, A. Hoyland, Models for variable-stress accelerated life testing experiments based on wiener processes and the inverse gaussian distribution, Theory Probab. Appl. 37 (1) (1993) 137–139, http://dx.doi.org/10.1080/00401706.1992.10485235.

[15] Z. Huang, Z. Xu, X. Ke, W. Wang, Y. Sun, Remaining useful life prediction for an adaptive skew-Wiener process model, Mech. Syst. Signal Process. 87 (2017) 294–306, http://dx.doi.org/10.1016/j.ymssp.2016.10.027.

[16] H. Cheng, X. Kong, G. Chen, Q. Wang, R. Wang, Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors, Measurement (2020) 108286, http://dx.doi.org/10.1016/j.measurement.2020.108286.

[17] M. Zhao, B. Tang, Q. Tan, Bearing remaining useful life estimation based on time–frequency representation and supervised dimensionality reduction, Measurement 86 (2016) 41–55, http://dx.doi.org/10.1016/j.measurement.2015.11.047.

[18] S. Zhao, Y. Zhang, S. Wang, B. Zhou, C. Cheng, A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method, Measurement 146 (2019) 279–288, http://dx.doi.org/10.1016/j.measurement.2019.06.004.

[19] S. Liu, Y. Hu, C. Li, H. Lu, H. Zhang, Machinery condition prediction based on wavelet and support vector machine, J. Intell. Manuf. 28 (4) (2017) 1045–1055, http://dx.doi.org/10.1007/s10845-015-1045-5.

[20] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, N. Zerhouni, Direct remaining useful life estimation based on support vector regression, IEEE Trans. Ind. Electron. 64 (3) (2016) 2276–2285, http://dx.doi.org/10.1109/TIE.2016.2623260.

[21] 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, http://dx.doi.org/10.1016/j.ymssp.2017.06.022.

[22] Z. Chen, A. Mauricio, W. Li, K. Gryllias, A deep learning method for bearing fault diagnosis based on cyclic spectral coherence and convolutional neural networks, Mech. Syst. Signal Process. 140 (2020) 106683, http://dx.doi.org/10.1016/j.ymssp.2020.106683.

[23] J. Zhu, N. Chen, W. Peng, Estimation of bearing remaining useful life based on multiscale convolutional neural network, IEEE Trans. Ind. Electron. 66 (4) (2018) 3208–3216, http://dx.doi.org/10.1109/TIE.2018.2844856.

[24] B. Wang, Y. Lei, N. Li, T. Yan, Deep separable convolutional network for remaining useful life prediction of machinery, Mech. Syst. Signal Process. 134 (2019) 106330, http://dx.doi.org/10.1016/j.ymssp.2019.106330.

[25] J. Chen, H. Jing, Y. Chang, Q. Liu, Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process, Reliab. Eng. Syst. Saf. 185 (2019) 372–382, http://dx.doi.org/10.1016/j.ress.2019.01.006.

[26] 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, http://dx.doi.org/10.1016/j.neucom.2017.02.045.

[27] L. Ren, X. Cheng, X. Wang, J. Cui, L. Zhang, Multi-scale dense gate recurrent unit networks for bearing remaining useful life prediction, Future Gener. Comput. Syst. 94 (2019) 601–609, http://dx.doi.org/10.1016/j.future.2018.12.009.

[28] Y. Chen, G. Peng, Z. Zhu, S. Li, A novel deep learning method based on attention mechanism for bearing remaining useful life prediction, Appl. Soft Comput. 86 (2020) 105919, http://dx.doi.org/10.1016/j.asoc.2019.105919.

[29] E. Sutrisno, H. Oh, A.S.S. Vasan, M. Pecht, Estimation of remaining useful life of ball bearings using data driven methodologies, in: 2012 IEEE Conference on Prognostics and Health Management, IEEE, 2012, pp. 1–7, http://dx.doi.org/10.1109/ICPHM.2012.6299548.

[30] 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, http://dx.doi.org/10.1016/j.procs.2018.01.106.

[31] 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, http://dx.doi.org/10.1016/j.dsp.2013.12.010.

Further reading

[1] M. Younan, E.H. Houssein, M. Elhoseny, A.A. Ali, Challenges and recommended technologies for the industrial internet of things: A comprehensive review, Measurement 151 (2020) 107198, http://dx.doi.org/10.1016/j.measurement.2019.107198.

[2] J.M. van Noortwijk, A survey of the application of gamma processes in maintenance, Eng. Syst. Saf 94 (1) (2009) 2–21.

[3] S.P. Kuniewski, J.A. van der Weide, J.M. van Noortwijk, Sampling inspection for the evaluation of time-dependent reliability of deteriorating systems under imperfect defect detection, Eng. Syst. Saf 94 (9) (2009) 1480–1490, http://dx.doi.org/10.1016/j.ress.2008.11.013.