Applied Soft Computing 129 (2022) 109604

Contents lists available at ScienceDirect

Applied Soft Computing

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

Feature Fusion based Ensemble Method for remaining useful life prediction of machinery

Gang Wang [a,b,c,*], Hui Li [a], Feng Zhang [a], Zhangjun Wu [a,b,c]

[a] School of Management, Hefei University of Technology, Hefei, Anhui, PR China
[b] Key Laboratory of Process Optimization and Intelligent Decision-making (Hefei University of Technology), Ministry of Education, Hefei, Anhui, PR China
[c] Ministry of Education Engineering Research Center for Intelligent Decision-Making & Information System Technologies, Hefei 230009, China

ARTICLE INFO

Article history:
Received 10 December 2021
Received in revised form 20 August 2022
Accepted 22 August 2022
Available online 5 September 2022

Keywords:
Remaining useful life
Feature fusion
Ensemble learning
Sparse learning
Bidirectional long short-term memory networks

ABSTRACT

The signal analysis features and deep representation features have been widely utilized to predict the Remaining Useful Life (RUL) of machinery. However, existing studies rarely fuse these features for RUL prediction to explore their complementarity. Therefore, this paper proposes a Feature Fusion based Ensemble Method (FFEM) that makes full use of the characteristics of signal analysis features and deep representation features. First, features are extracted by signal analysis and deep learning methods, respectively. The time-domain features, frequency-domain features, and time–frequency domain features are extracted by different signal analysis methods, while the deep representation features are from the bidirectional long short-term memory networks. Then, an improved random subspace method is proposed, which fuses different types of features based on group-based sparse learning to use the complementarity among features. Furthermore, accurate and diverse base learners are generated and the aggregation strategy, mean rule, is adopted for predicting RUL. To validate the proposed FFEM, experiments on the run-to-failure datasets of bearings are conducted, and the experimental results verify that the proposed method greatly improves the RUL prediction performance and surpasses other existing ensemble learning methods.

1. Introduction

Along with the growing complexity of the machinery, conventional maintenance strategies including breakdown maintenance and preventive maintenance are incapable of meeting the reliability and safety requirements of industrial equipment [1]. Consequently, Prognostics and Health Management (PHM) has attracted increasing attention in the past years [2–5]. As an indispensable and valuable aspect of PHM, Remaining Useful Life (RUL) prediction focuses on estimating the time of machinery failure. Predicting RUL accurately can support the maintenance decision-making to prevent the machine from catastrophic failures to some extent and reduce maintenance costs [6]. In recent years, RUL prediction become crucially important in both the modern industry and academic research fields.

The RUL prediction methods can be categorized as the model-based method [7], statistical-based method [8], and machine learning-based method [9]. With the improvement of sensor techniques and signal transmission techniques, the machine learning-based method has drawn more and more attention, which can directly model the relationship between historical data and the RUL of machinery. Unlike model-based and statistical-based methods, the machine learning-based method can predict RUL without prior degradation knowledge [10]. In general, there are two important steps of the machine learning-based method, feature extraction and prediction model construction [11].

As for the feature extraction, features can be extracted by the signal analysis and deep learning methods. Signal analysis methods, such as the time-domain method, frequency-domain method, and time–frequency domain method have been widely used in the field of RUL prediction, which can extract low-level intuitive features [12]. The time-domain method calculates statistics from the signal in the time domain, which can reflect the whole degradation tendency of machinery [13]. The frequency-domain method is based upon the transformed signal over the frequency domain, which can distinguish and detach specific frequency components of interest [14]. Rather than concentrating on the time domain or frequency domain alone, the time–frequency domain method can identify time-dependent variations of frequency components in the signal, enabling it to analyze non-stationary signals [13,15,16]. Recently, deep learning methods including Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) have achieved success in areas including

* Corresponding author at: School of Management, Hefei University of Technology, Hefei, Anhui 230009, PR China
E-mail address: wgedison@hfut.edu.cn (G. Wang).

https://doi.org/10.1016/j.asoc.2022.109604
1568-4946/

image recognition and speech recognition, which also attracted attention in the PHM field [17–20]. Using multilayer neural networks, deep learning methods can learn high-level abstract representations from the signal input automatically and accurately. Furthermore, signals obtained from sensors are time series. If ignoring the temporal characteristic in signals, the extracted features may lack key information, causing poor performance of RUL prediction [20]. Fortunately, Recurrent Neural Network (RNN) and its variants such as LSTM by introducing self-feedback neurons into the network structure can deal with sequential data, which have shown great ability on RUL prediction [16,21]. In conclusion, low-level intuitive information and high-level abstract information in the above features are both effective on RUL prediction. Different features represent distinct characteristics of the raw data and naturally correspond with group structure. However, existing studies rarely fuse these features to explore their complementarity in RUL prediction.

As for the prediction model construction, machine learning methods have been widely utilized in RUL prediction, examples are Extreme Learning Machine (ELM), Multi-Layer Perceptron (MLP), and Support Vector Regression (SVR) [22–25]. However, in light of the variability of machinery working conditions and the complexity of modern industrial systems, the single model suffers from issues of instability and low generalization ability [26]. In this regard, ensemble methods including Bagging and Boosting have been employed to construct prediction models in the field of RUL prediction [27,28]. By training diverse base learners and combining them through certain strategies, the ensemble method has the potential to solve complicated issues and achieve more robust and accurate RUL prediction performance [29]. Note that the signal analysis and deep representation features may bring about high-dimensional problems. Compared with Bagging and Boosting, the Random Subspace (RS) may be more potential to handle data with high dimensions as it trains base learners on feature subspaces instead of the whole feature space [30]. Nevertheless, the standard RS generates feature subspaces by the complete randomness of feature sampling and lacks the mechanism of effectively utilizing the complementarity among different features. The base learners may still be affected by the negative influence of irrelevant or redundant features, leading to poor prediction performance and generalization ability.

Accordingly, a novel Feature Fusion based Ensemble Method (FFEM) is proposed to explore the complementarity among different features for improving the RUL prediction performance in this paper. First, features with different characteristics are extracted by signal analysis and deep learning methods, respectively. Then, the improved RS ensemble model for predicting RUL is constructed. Concretely, the group-based sparse learning model takes into account the group structure among features, making it possible for RS to effectively utilize the complementarity among distinct features and restrain the negative impact of irrelevant and redundant features. Finally, accurate and diverse base learners are generated, and the aggregation strategy, mean rule, is adopted for RUL prediction. To validate the performance of the proposed ensemble method, experiments are conducted using the run-to-failure datasets of bearings from the PRONOSTIA platform [31]. The experimental results verified the superior performance of the proposed FFEM compared to other commonly used ensemble methods. Furthermore, the predictive performances of every group feature and the fusion features from signal analysis features and deep representation features are also discussed. Results illustrated that FFEM is an effective method for the RUL prediction of bearings, and has the potential to deal with RUL prediction problems of other objects.

The main contributions of this paper are summarized as follows:

(1) An improved framework for enhancing the performance of RUL prediction is proposed. In this framework, the complementarity among the grouping features can be effectively utilized to improve the RUL prediction performance.

(2) A novel RUL prediction method, FFEM, is presented. By introducing group-based sparse learning to enhance the RS, the negative influence of irrelevant and redundant features is reduced, and signal analysis features and deep representation features can be fused properly.

(3) The proposed FFEM is validated on the bearing run-to-failure datasets to predict RUL. The experiments verified the effectiveness and superiority of FFEM compared to other commonly used machine learning methods.

The remainder of this paper is organized as follows. The related works about machine learning-based methods of RUL prediction are reviewed in Section 2. The proposed FFEM method is detailedly described in Section 3. The experimental design part comprising the dataset, evaluation metrics, and experimental procedures is presented in Section 4. Then, in Section 5, the experimental results of the proposed FFEM and other methods are reported and analyzed. Finally, Section 6 summarizes this paper and presents prospects for future work.

2. Related work

RUL prediction is a critical and challenging problem in PHM. With abundant raw signal data, numerous studies have applied machine learning to predict RUL over the past years. The performance of the machine learning-based RUL prediction method mainly depends on the quality of the extracted features and the effectiveness of the constructed prediction model [32].

2.1. Feature extraction

The signal analysis and deep learning methods can be utilized to extract signal analysis features and deep representation features. The signal analysis features represent the low-level intuitive information of the raw signal data. Different domain features extracted by the time-domain method, frequency-domain method, and time–frequency domain method have been widely used for RUL prediction [11,25,33]. For example, Atamuradov et al. extracted time-domain features such as crest factor, kurtosis, mean, skewness, peak-to-peak, root mean square, standard deviation, and variance for RUL prediction, and the effectiveness was verified using data of switch machine [34]. Xia et al. applied the Fast Fourier Transform (FFT) to extract frequency-domain features, and the prediction performance was verified using the bearing degradation datasets under different working conditions [35]. Benkedjouh et al. extracted time–frequency features from raw signals by using Wavelet Packet Decomposition (WPD), which could describe the evolution of the bearing degradation and were effective to predict the RUL [32].

Recently, deep learning methods are receiving increasing attention in the field of RUL prediction, since they can extract high-level abstract feature representations from the raw signal data automatically and accurately. For instance, Li et al. utilized Convolutional Neural Network (CNN) to predict the RUL of aero-engine units, in which the collected raw data is normalized and then input into the network for feature extraction [36]. Ma et al. employed stacked Sparse Autoencoder (SAE) to extract degradation features from various sensors automatically, which could improve the performance of the RUL prediction [37]. Deutsch and He achieved accurate RUL prediction based on Deep Belief Network (DBN), which made use of the self-taught feature learning ability of the DBN to extract deep representation features [38]. Although, signals obtained from sensors are time series

in essence. If ignoring the temporal characteristics in signals, it may lose some critical information in the process of feature learning, resulting in poor performance of RUL prediction. In this situation, RNN and its variants can deal with sequential data by introducing self-feedback neurons into the network structure and have been introduced to the field of RUL prediction. For instance, Yu et al. trained a bidirectional RNN to convert high-dimensional signals to low-dimensional features for RUL prediction, and the experiment results on the milling machine demonstrated the competitiveness of these features [39]. Al-Dulaimi et al. employed LSTM to extract temporal features, and their effectiveness for RUL estimation was verified on the engine dataset [40]. Zhao et al. employed bidirectional GRU to extract deep representation features for tool wear prediction, which captured various health degradation patterns [41].

In summary, both signal analysis features and deep representation features are useful for RUL prediction. Different features represent distinct degradation information and naturally correspond with group structure. However, the existing research ignores the complementarity among these features and rarely fuses them for RUL prediction. Besides, the direct fusion of different features without distinguishing their group properties may weaken the quality of the features. Therefore, a better fusion mechanism should be designed to utilize these features complementarily and improve the performance of RUL prediction.

2.2. Prediction model construction

In the phase of prediction model construction, without prior degradation knowledge, many machine learning methods including ELM, MLP, and SVR have been widely used to construct prediction models [23,42–44]. For example, Javed et al. employed the ELM for model construction to predict RUL, and the effectiveness of ELM was validated on cutting tools datasets [45]. Huang et al. utilized the MLP to construct a prediction model for estimating RUL values of the bearings, whose prediction performance surpassed the L10 bearing life prediction formula that is widely utilized by manufacturers [46]. Besides, due to outstanding generalization performance attributed to the structural risk minimization principle, the SVR has been successfully applied in RUL prediction problems [43]. For instance, Khelif et al. used the signal data directly to predict the RUL of the turbofan, and the results of the experiment indicated that the SVR performed better than ANN for RUL prediction [47]. In [25], the SVR was used to predict the RUL of bearings, and the experimental results demonstrated that SVR was superior to ANN, LSTM, and autoregressive moving average model, etc.

However, since the working conditions of machinery are becoming more variable and the modern industrial systems are more complicated, the above methods suffer from the issues such as instability and low generalization ability. In this situation, ensemble learning methods train base learners and combine them through a certain strategy for prediction, with the power to deal with the above problems [29]. For example, Rigamonti et al. adopted the Bagging and Echo State Networks (ESNs) with different architectures to realize the diversity of the model, and the experiment on real datasets showed that the performance of the ensemble learning model surpassed individual ESN [48]. Sun et al. used Boosting to train several MLP neural networks, and the experiments on the turbofan engine showed that the ensemble methods for the RUL estimation attained higher performance [49]. Wu et al. applied Random Forests (RF) to construct a prediction model and the experiment on tool wear datasets verified the effectiveness of combining a set of regression trees [50].

As aforementioned, the signal analysis and deep representation features can be fused to predict RUL, but these features may lead to data with the problem of high dimension. Compared with the Bagging and the Boosting, RS is more potential for dealing with this problem because it trains base learners on feature subspaces rather than the whole feature space. However, the standard RS generates feature subspaces by random feature sampling and lacks the mechanism of effectively utilizing the complementarity among features. The base learners may still be affected by the deleterious influence of irrelevant and redundant features, leading to poor prediction performance and generalization ability.

3. Proposed method

3.1. Framework of FFEM

RUL prediction is very critical since it contributes to preventing the machine from catastrophic failures to some extent and reducing maintenance costs [51]. However, previous studies rarely explored the effect of fusing signal analysis features and deep representation features, and it is also a challenge to effectively utilize the complementarity among these features. Therefore, a novel feature fusion based ensemble method is proposed in this paper to integrate the above features and improve the RUL prediction performance. As illustrated in Fig. 1, the framework of FFEM mainly includes three steps:

(1) Data acquisition: During the machining process, the raw signal data relating to RUL are collected from sensors. Especially, the run-to-failure signal data under constant operating conditions can be acquired. Correspondingly, the RUL of the instances can be labeled according to the total life of machinery and the recording time.

(2) Feature extraction: The signal analysis methods including the time-domain method, frequency-domain method, and time–frequency domain method, are utilized to extract signal analysis features. At the same time, bidirectional LSTM (BiLSTM) is utilized to extract deep representation features to capture temporal information in the raw signal data.

(3) Prediction model construction: The improved RS ensemble model is enhanced by the group-based sparse learning, where the group structure among features is considered and high-quality features can be identified, making it possible for RS to use the complementarity among features. Then, accurate and diverse base learners are generated, and the mean rule is adopted as the aggregation strategy for RUL prediction.

For clarity of description, some basic notations utilized in this section are presented as follows: Given training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, it is the raw signal data of the whole life with $N \in \mathbb{R}$ historical units. $\mathbf{x}_i \in \mathbb{R}^M$ is a time-series signal of length M, and y_i is the true RUL value corresponding to $\mathbf{x}_i$. The goal is to predict RUL accurately for new instances by extracting features from the raw signal and constructing the best model $y_i = \mathrm{f}(\mathbf{x}_i)$, which determines the mapping relationship between the raw signal data and the values of RUL.

3.2. Feature extraction

3.2.1. Signal analysis features extracted by different domains

Features have an important influence on the RUL prediction, and good features should characterize the health status of the machinery. In this paper, signal analysis features are extracted by three signal analysis methods including the time-domain, frequency-domain, and time–frequency domain methods.

Studies have validated that time-domain features can reflect the degradation evolution of the run-to-failure machinery [52]. To be specific, some time-domain features can reflect the energy and amplitude in the time domain of the signal while others

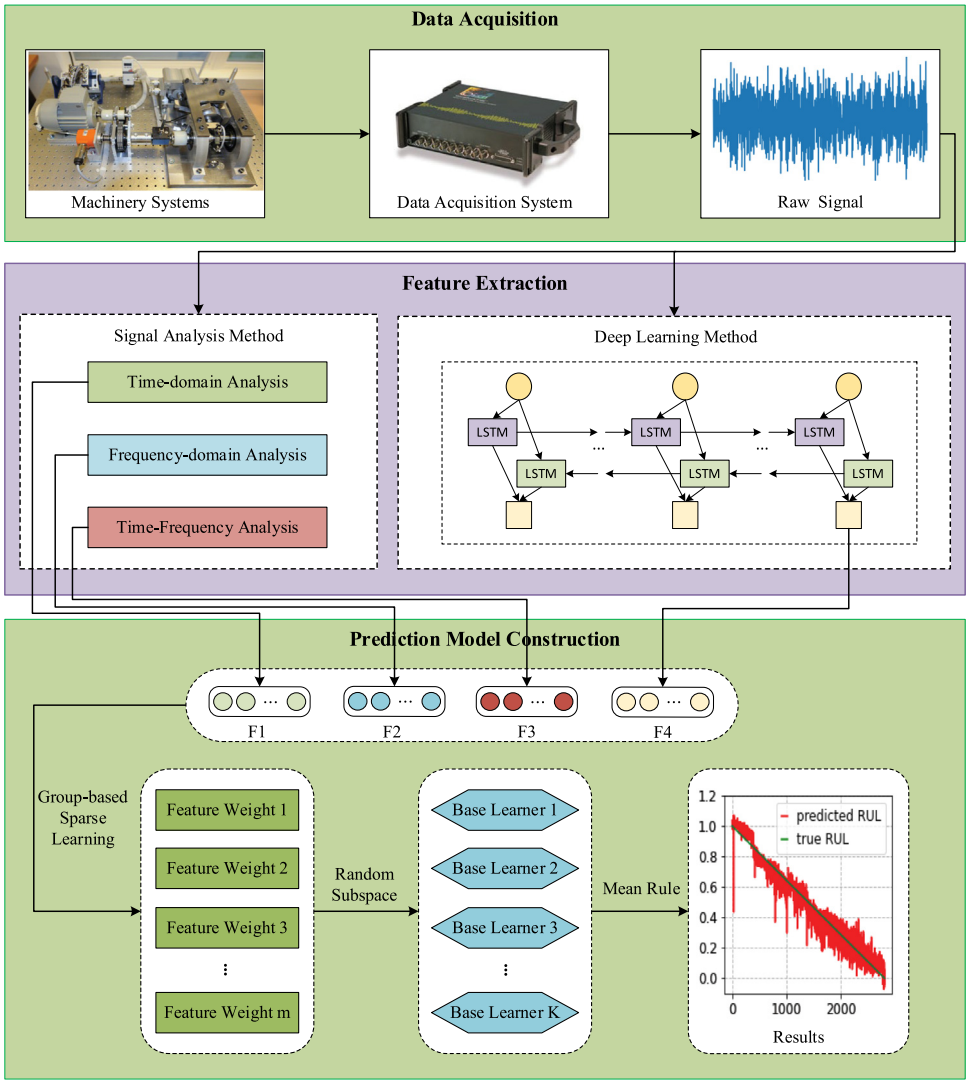

Fig. 1. Framework for RUL prediction.

reflect the distribution characteristics in the time domain [53]. For the time-series signal x_i, the calculation formulas of the extracted features are depicted in Table 1, where x_{ij} stands for the elements in each time vector $\mathbf{x}_i$. Overall, 16 features involve commonly used time-domain features, including minimum (X_{min}), maximum (X_{max}), mean (X_{mean}), square root of amplitude (X_{sra}), root variance (X_{rv}), mean absolute value (X_{mav}), skewness value (X_{sv}), root mean square (X_{rms}), kurtosis value (X_{kv}), margin factor (X_{mf}), kurtosis factor (X_{kf}), skewness factor (X_{skf}), impulse factor (X_{if}), shape factor (X_{shf}), crest factor (X_{cf}), and peak-peak value (X_{ppv}) [54]. The time-domain features can be represented as $\mathrm{F}_1 = \{\mathrm{F}_1^1, \mathrm{F}_2^1, \mathrm{F}_3^1, \ldots, \mathrm{F}_{m_1}^1\}$, where $m_1 = 16$.

Frequency-domain features can display how much of the signal exists in a given frequency band, and identify valuable information that may not be found in the time domain [55]. Among frequency-domain analysis methods, such as cepstrum, envelope analysis, Fast Fourier Transform (FFT), higher-order spectra, and spectral analysis, the most widely used is the FFT [56]. FFT is a simple and well-established method, which can transform a time series $\mathbf{x}_i$ into its frequency coefficients. The absolute values $\boldsymbol{\beta}_i \in \mathbb{R}^L$ of these coefficients indicate the intensity of the corresponding frequency components $\mathbf{f}_i \in \mathbb{R}^L$ in the time-series signal [57]. In this paper, the frequency-domain features are calculated through the $\boldsymbol{\beta}_i$ and $\mathbf{f}_i$, and L is set as $M/2$. Details of the frequency domain features, such as the root variance of frequency (X_{rvf}) and mean of frequency (X_{meanf}), are shown in Table 1. The frequency-domain features can be represented as $\mathrm{F}_2 = \{\mathrm{F}_1^2, \mathrm{F}_2^2, \mathrm{F}_3^2, \ldots, \mathrm{F}_{m_2}^2\}$, where $m_2 = 12$.

Time–frequency methods including the Short-Time Fourier Transform (STFT), the Wavelet Transform (WT), and Hilbert–Huang Transform (HHT), etc., are proposed for non-stationary signals [58–60]. The advantage of the time–frequency method is its capability of analyzing given signals from the point of view of both the time domain and the frequency domain. However, limited by the time–frequency resolution capability, it is hard to distinguish the low frequencies within short windows using STFT [55]. Although WT provides richer pictures than STFT, it only re-decomposes low-frequency signals, making its resolution decrease with increasing frequency. As an enhancement of WT, the Wavelet Package Transforms (WPT) overcomes this drawback, which can divide a signal into low-frequency and high-frequency bands in the form of a binary tree [61]. Through d-level WPT, a total of 2^d sub-bands can be generated in the last level. Next, the node energies of the final 2^d wavelet packets can be figured out and then normalized as the features of the wavelet packets [30, 62]. In this paper, by decomposing the raw signal in the $5th$ level with the mother wavelet 'DB4', 2^5 wavelet packet features are calculated as time–frequency domain features ultimately. The

Table 1
Details of the time-domain features and frequency-domain features.

Time-domain Features		Frequency-domain Features	
$X_{max} = \max(\mathbf{x}_i)$	$X_{min} = \min(\mathbf{x}_i)$	$X_{maxf} = \max(\boldsymbol{\beta}_i)$	$X_{minf} = \min(\boldsymbol{\beta}_i)$
$X_{sra} = \left(\frac{1}{M}\sum_{j=1}^{M}\sqrt{\lvert x_{ij}\rvert}\right)^2$	$X_{mean} = \frac{1}{M}\sum_{j=1}^{M} x_{ij}$	$X_{rmsf} = \left(\frac{1}{L}\sum_{l=l}^{L}\beta_{il}^2\right)^{\frac{1}{2}}$	$X_{meanf} = \frac{1}{L}\sum_{l=1}^{L}\beta_{il}$
$X_{rv} = \left(\frac{1}{M}\sum_{j=1}^{M}(x_{ij}-X_{mean})^2\right)^{\frac{1}{2}}$	$X_{mav} = \frac{1}{M}\sum_{j=1}^{M}\lvert x_{ij}\rvert$	$X_{rvf} = \left(\frac{1}{L}\sum_{l=1}^{L}(\beta_{il}-X_{meanf})^2\right)^{\frac{1}{2}}$	$X_{fc} = \frac{1}{L}\sum_{l=1}^{L}\frac{(f_{il}\cdot\beta_{il})}{X_{meanf}}$
$X_{sv} = \left(\frac{1}{M}\sum_{j=1}^{M}\frac{(x_{ij}-X_{mean})}{X_{rv}}\right)^3$	$X_{rms} = \left(\frac{1}{M}\sum_{j=1}^{M}x_{ij}^2\right)^{\frac{1}{2}}$	$X_{svf} = \left(\frac{1}{L}\sum_{l=1}^{L}\left(\frac{\beta_{il}-X_{meanf}}{X_{ryf}}\right)\right)^3$	$\left(\frac{1}{L}\sum_{l=1}^{L}(f_{il}-X_{fc})^2\cdot\beta_{il}\right)^{\frac{1}{2}}$
$X_{kv} = \left(\frac{1}{M}\sum_{j=1}^{M}\frac{(x_{ij}-X_{mean})}{X_{rv}}\right)^4$	$X_{mf} = \frac{\max(\lvert\mathbf{x}_i\rvert)}{X_{sra}}$	$X_{kvf} = \left(\frac{1}{L}\sum_{l=1}^{L}\frac{(\beta_{il}-X_{meanf})}{X_{rvf}}\right)^4$	$\left(\frac{1}{L}\sum_{l=1}^{L}\frac{(f_{il}^2\cdot\beta_{il})}{X_{meanf}}\right)^{\frac{1}{2}}$
$X_{kf} = X_{kv}/X_{rms}^4$	$X_{skf} = X_{kv}/X_{rv}^3$	$X_{kff} = X_{kvf}/X_{rmsf}^4$	$X_{skff} = X_{kvf}/X_{rvf}^3$
$X_{if} = \max(\lvert\mathbf{x}_i\rvert)/X_{mav}$	$X_{shf} = X_{rv}/X_{mav}$		
$X_{cf} = \max(\lvert\mathbf{x}_i\rvert)/X_{rms}$	$X_{ppv} = X_{\max} - X_{\min}$		

time–frequency domain features can be represented as $F_3 = \{F_1^3, F_2^3, F_3^3, \ldots, F_{m_3}^3\}$, where $m_3 = 32$.

3.2.2. Deep representation features extracted by BiLSTM

The above signal analysis features only reflect the low-level intuitive information of the raw signal data, while deep learning methods are capable of extracting high-level abstract representations. Moreover, RUL prediction is a typical sequential problem, and the raw signal data obtained by the sensors during the whole life of machinery is time series essentially [63]. The RNNs have been proven to have the capacity to analyze hidden sequential patterns in time series. However, when the data volume is large, the RNNs may face the "gradient vanishing" problem, which means that RNNs can forget the earlier inputs of the sequence in the case of long-term sequences. LSTM can overcome the problem of gradient vanishing, which can capture the long-term dependence by memory cell and gate units including input, output, and forget gates [64]. Nevertheless, LSTM just captures the dependent information of the current state on the previous state. The lack of future information may bring about an inadequate understanding of the input. If both the past and future information are considered, sequence learning tasks can be facilitated [65]. The BiLSTM can simultaneously learn forward and backward temporal information of time-series signals, contributing to reflecting the degradation process of the machinery. Accordingly, in this paper, BiLSTM is adopted to extract deep representation features.

A time-series signal is defined as one sequence input $\mathbf{x} \in \mathbb{R}^M$, where M denotes the size of the sequence length. Divide each sequence of size M into $\mathbf{x}' = [\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T]^T$ using the non-overlapping sliding time windows method, where $\mathbf{x}_t \in \mathbb{R}^d$, d is the size of time windows and $M = d * T$. The core idea of BiLSTM is that using two separate LSTM layers to process the sequence in two directions, forward and backward process. For each $\mathbf{x}_t$, the forward layer encodes $\mathbf{x}_t$ by gathering the past information from $\mathbf{x}_1$ to $\mathbf{x}_t$ for forward hidden state $\overrightarrow{\mathbf{h}}_t$. Similarly, the backward layer encodes $\mathbf{x}_t$ based on future information from $\mathbf{x}_T$ to $\mathbf{x}_t$ for the backward hidden state $\overleftarrow{\mathbf{h}}_t$.

The LSTM architecture is constituted of a group of recurrently connected subnets, which can be called memory blocks. At each time step, there is an LSTM memory block in the forward hidden layer and the backward hidden layer. Fig. 2 illustrates an LSTM memory block with a single cell. In the LSTM memory block at each time step, the current hidden vector $\mathbf{h}_t$ is computed based on the past hidden vector $\mathbf{h}_{t-1}$, the previous cell vector $\mathbf{c}_{t-1}$, and the current input $\mathbf{x}_t$. The above process can be shortly represented

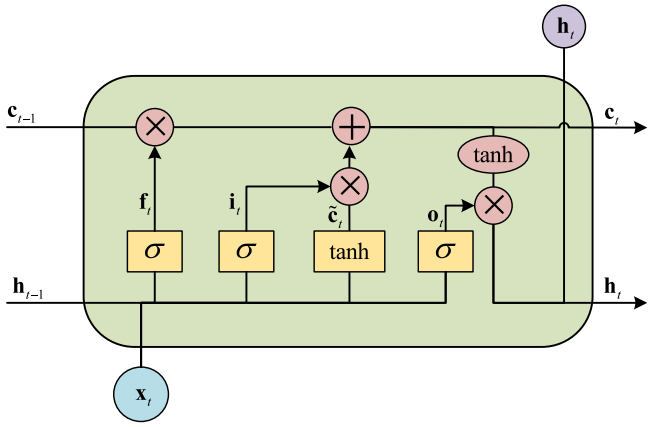

Fig. 2. The LSTM memory block.

by $\mathbf{h}_t = \text{lstm}(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1})$. The details of the operation in the LSTM memory block can be defined in the following [64]:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i^x \mathbf{x}_t + \mathbf{W}_i^h \mathbf{h}_{t-1} + \mathbf{b}_i) \tag{1}$$

where $\mathbf{i}_t$ denotes input gate, σ is the logistic sigmoid function, $\mathbf{W}_i^x$ denotes weight matrix of the input vector $\mathbf{x}_t$, $\mathbf{W}_i^h$ denotes weight matrix of hidden vector $\mathbf{h}_{t-1}$, and $\mathbf{b}_i$ denotes the bias vector of the input gate. The decision upon taking which information of the short-term state $\mathbf{h}_{t-1}$ and the current input $\mathbf{x}_t$ into account to update the current cell state is made by the input gate.

In the meanwhile, the forget gate $\mathbf{f}_t$ decides on which information is abandoned from the memory cell:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f^x \mathbf{x}_t + \mathbf{W}_f^h \mathbf{h}_{t-1} + \mathbf{b}_f) \tag{2}$$

where $\mathbf{W}_f^x$ and $\mathbf{W}_f^h$ denote the weight matrices of $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$ separately, and $\mathbf{b}_f$ denotes the bias vector of the forget gate.

Then, the current memory cell $\mathbf{c}_t$ is updated by the input gate $\mathbf{i}_t$ and forget gate $\mathbf{f}_t$.

$$\widetilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c^x \mathbf{x}_t + \mathbf{W}_c^h \mathbf{h}_{t-1} + \mathbf{b}_c) \tag{3}$$

$$\mathbf{c}_t = \widetilde{\mathbf{c}}_t \odot \mathbf{i}_t + \mathbf{c}_{t-1} \odot \mathbf{f}_t \tag{4}$$

where $\widetilde{\mathbf{c}}_t$ refers to the new cell state candidate, tanh is the hyperbolic tangent function, $\mathbf{W}_c^x$ and $\mathbf{W}_c^h$ denote the weight matrices of $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$ separately, and $\mathbf{b}_c$ denotes the bias vector, $\odot$ is the element-wise multiplication.

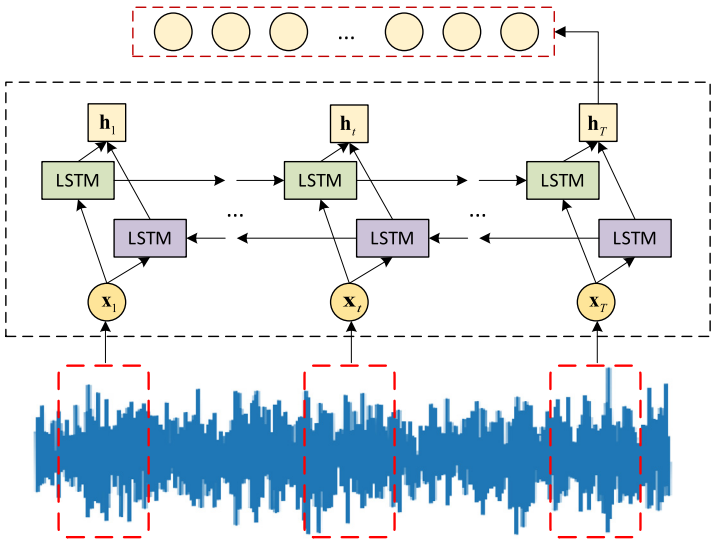

Fig. 3. The BiLSTM network.

Similar to the mechanisms of the input gate and forget gate, the output gate $\mathbf{o}_t$ can be obtained by Eq. (5), which determines which information of the cell states are output.

$$\mathbf{o}_t = \sigma(\mathbf{W}_o^x \mathbf{x}_t + \mathbf{W}_o^h \mathbf{h}_{t-1} + \mathbf{b}_o) \tag{5}$$

where $\mathbf{W}_o^x$ and $\mathbf{W}_o^h$ denote the weight matrices of $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$, and $\mathbf{b}_o$ denotes the bias vector of the output gate.

The LSTM unit's output $\mathbf{h}_t$ is obtained from the current memory cell $\mathbf{c}_t$ and the output gate $\mathbf{o}_t$.

$$\mathbf{h}_t = \tanh(\mathbf{c}_t) \odot \mathbf{o}_t \tag{6}$$

The BiLSTM network for extracting deep representation features is shown in Fig. 3. To extract deep representation features with better prediction performance, the linear regression layer is stacked upon the BiLSTM network in this paper. In the training of the deep learning networks, the Mean Squared Error (MSE) function is utilized as the cost function. The hyperparameters of the BiLSTM are chosen by grid search method and hold-out validation. The parameters are optimized by Adaptive moment estimation (Adam) [66] through the Back Propagation Through Time (BPTT) algorithm [67]. Finally, $\overrightarrow{\mathbf{h}}_t = \text{lstm}\left(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}, \overrightarrow{\mathbf{c}}_{t-1}\right)$ and $\overleftarrow{\mathbf{h}}_t = \text{lstm}\left(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}, \overleftarrow{\mathbf{c}}_{t+1}\right)$ at the last time step can be obtained from the forward layer and backward layer. Concatenate them together as the deep representation feature $\mathbf{h}_t = \left[\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t\right]$. The deep representation features can be represented as $\mathrm{F}_4 = \{\mathrm{F}_1^4, \mathrm{F}_2^4, \mathrm{F}_3^4, \ldots, \mathrm{F}_{m_4}^4\}$, where $m_4 = 128$.

3.3. Prediction model construction

Some ensemble methods have been employed in the area of RUL prediction to enhance the generalization ability and accuracy of the model. According to the way of the base learners generation, ensemble methods are categorized as instance partitioning methods or feature partitioning methods [68]. Specifically, Bagging and Boosting both belong to the instance partitioning ensemble methods. The Bagging generates base learners by using various bootstrapped instances in the training dataset, and the Boosting sequentially reweights the instances to obtain diverse base learners [69–71]. The feature partitioning ensemble methods such as RS train base learners in feature subspaces instead of the whole feature space [72]. Note that the complementarity among different features can be explored to enhance the RUL prediction performance, but these features may result in high-dimensional problems. Hence, RS may be more suitable to construct the prediction model. However, the standard RS generates feature subspaces by the random feature sampling, which could neither fully exploit multiple features nor remove the low-quality features, thus generating poor base learners. In this situation, it's essential to distinguish the relative importance of each feature to reduce the probability of irrelevant and redundant features being sampled.

The regularized sparse methods can be used as they assign weights to features. For the regularized sparse methods, Least Absolute Shrinkage and Selection Operator (LASSO) is a typical representative and has been widely used to pick out important features [73]. However, LASSO neglects the distinct properties of different groups. Subsequently, Group LASSO (GLASSO) is proposed as an improvement of LASSO by keeping or eliminating homogeneous features simultaneously [74]. The GLASSO does not, however, yield sparsity within a feature group. That is, if a group of parameters is non-zero, all features in that group will all be non-zero, meaning that the difference among the features in a group is ignored. By combining the main principles of LASSO and GLASSO, the Sparse GLASSO (SGL) implements sparsity both on the group level and within-group level [75]. After estimating feature weights, particularly important features can be identified.

Therefore, an improved RS ensemble model is constructed, in which the RS is enhanced by SGL and different types of features can be well fused to explore the complementarity among them. Specifically, SGL can identify not only the important feature groups but also the important individual features within each selected group. As shown in Fig. 4, by endowing feature groups with varied weight vectors, the relative importance of the feature groups as well as the individual features are acquired. Then, the different feature subspaces can be acquired by the improved RS, and diverse base learners can be trained using SVR in different subspaces. Finally, the mean rule is adopted to combine all RUL prediction results of base learners.

Firstly, SGL estimation is utilized to yield sparsity at both the group and individual feature levels for the input features. Given training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, we can transform it into the matrix $\mathbf{X} \in \mathbb{R}^{N*m}$ and vector $\mathbf{y} \in \mathbb{R}^N$, where the N denotes the number of training data instances and $m = (m_1 + m_2 + m_3 + m_4)$ is the

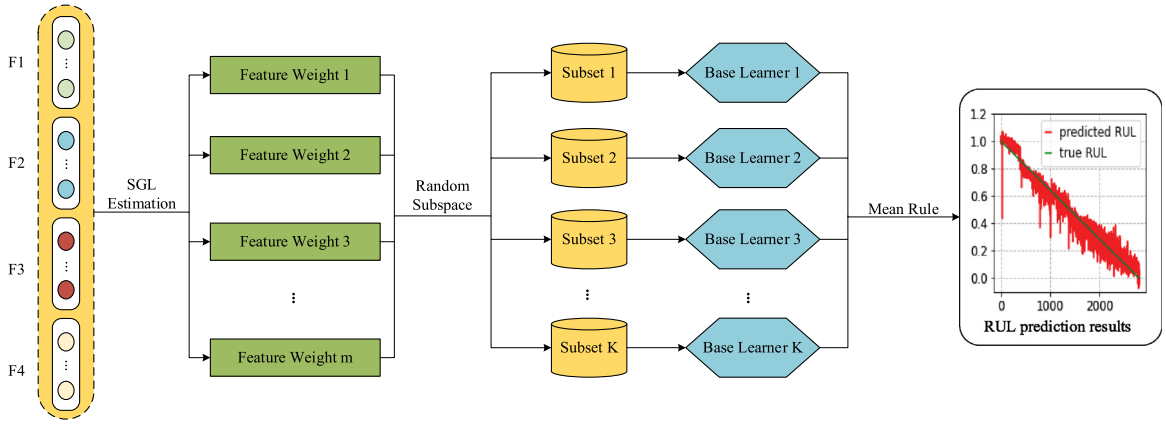

Fig. 4. The procedure of the improved RS ensemble model.

total number of all features. SGL is defined as a regularized linear regression as follows:

$$\min_{\boldsymbol{\omega}} \frac{1}{2n}\left\|\mathbf{y}-\sum_{g=1}^{4}\mathbf{X}^{(\mathrm{F}_g)}\boldsymbol{\omega}^{(\mathrm{F}_g)}\right\|_2^2+(1-\alpha)\lambda\Big(\sum_{g=1}^{4}\sqrt{m_g}\left\|\boldsymbol{\omega}^{(\mathrm{F}_g)}\right\|_2\Big) + \alpha\lambda\left\|\boldsymbol{\omega}\right\|_1 \tag{7}$$

where $\mathbf{X}^{(\mathrm{F}_g)}$ is the submatrix of $\mathbf{X}$ with columns corresponding to the feature group F_g, $\boldsymbol{\omega}^{(F_g)} = (\boldsymbol{\omega}_1^{(g)},\ldots,\boldsymbol{\omega}_{m_g}^{(g)})$ denotes the coefficient vector of that feature group, $\boldsymbol{\omega}_{m_g}^{(g)}$ denotes the individual coefficient concerning the feature $\mathrm{F}_{m_g}^{g}$. The first penalty term $(1-\alpha)\lambda(\sum_{g=1}^{4}\sqrt{m_g}\left\|\boldsymbol{\omega}^{(\mathrm{F}_g)}\right\|_2)$ places sparsity upon feature groups and the second penalty term $\alpha\lambda\left\|\boldsymbol{\omega}\right\|_1$ is designed to produce sparse effects for individual features, with $\|\bullet\|_q$ as the L_q norm. In detail, by employing both the "group-wise sparsity" and "within-group sparsity", SGL achieves the overall sparsity. The "group-wise sparsity" is referring to the number of feature groups with no less than one non-zero coefficient, and "within-group sparsity" refers to the number of non-zero coefficients within every non-zero feature group [75]. Besides, the parameter $\alpha \in [0, 1]$ is the key for SGL to combine the GLASSO and LASSO (SGL degenerates into the GLASSO or LASSO when $\alpha = 0$ or $\alpha = 1$). The parameter λ is utilized to regulate the level of sparsity for "group-wise sparsity" and"within-group sparsity". When λ tends to be very small, the weights of most features are prone to be non-zero values on account of its pointless constraint effects. Conversely, as setting λ to a significantly big value, the weights of irrelevant or redundant features could be shrunk into 0, so that they are hardly selected in the following feature sampling process [76]. In summary, a high-quality feature selection is dependent on both α and λ. After the SGL estimation, the sparse coefficient vector $\boldsymbol{\omega}$ corresponding to individual features is obtained, which determines the importance scores for each feature. Furthermore, the weight w_j' corresponding to the jth feature can be calculated by Eq. (8), which acts as the selection probability of the jth feature.

$$w_j' = \frac{|\omega_j|}{\sum_{j=1}^{m}|\omega_j|} \tag{8}$$

Then controlled by the feature weight $\mathbf{w}'$ and subspace rate *Ratio*, K feature subspaces $\{S^1, S^2, S^3, \ldots, S^K\}$ are generated from the original feature space. To be specific, after performing feature selection for every feature subspace iteratively, a set of data subsets $\{D^1, D^2, D^3, \ldots, D^K\}$ can be obtained. This way, the different types of features can be well fused into different feature subspaces, while the diversity among base learners is also guaranteed by randomly selecting the features. The accurate base learner is the basic requirement for ensemble methods. SVR is adopted as the base learner in this paper. As a promising tool for regression issues, SVR has been widely utilized for various problems of time series prediction, such as financial time series forecasting, wind speed prediction, and engine reliability prediction [77]. SVR has the advantage that computational complexity is not influenced by the dimensionality of the input space. Moreover, SVR demonstrates good generalization capability and high prediction performance [78]. Thus, this paper employs SVR as the base learner. Given a data subset of instances $D^k = \{(\mathbf{x}_i^k, y_i)\}_{i=1}^N$, where $\mathbf{x}_i^k \in \mathbb{R}^m$ denotes the input feature concerning the output y_i, and m represents the dimension size of the feature. The regression function estimated by SVR can be denoted as:

$$\mathrm{f}(\mathbf{x}) = \mathbf{w}\cdot\phi(\mathbf{x}) + b \tag{9}$$

where $\mathbf{w}$ denotes the weight vector, b denotes the bias term, and $\phi(\cdot)$ denotes the nonlinear mapping function. To obtain the optimal values of $\mathbf{w}$ and b, the regularized risk optimization problem (10) should be solved:

$$\min_{\mathbf{w},b,\zeta,\zeta^*}\frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{N}(\zeta_i+\zeta_i^*) \quad s.t.\begin{cases} y_i-\mathbf{w}^T\phi(\mathbf{x}_i^k)-b\le\varepsilon+\zeta_i, \\ \mathbf{w}^T\phi(\mathbf{x}_i^k)+b-y_i\le\varepsilon+\zeta_i^*, \\ \zeta_i,\zeta_i^*\ge 0, i=1,2,\ldots,N \end{cases} \tag{10}$$

where the objective function includes two terms: the regularization term $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ and the empirical risk $\sum_{i=1}^{N}(\zeta_i+\zeta_i^*)$, the penalty factor C determines the trade-off between the former term and the latter, ε is the allowable maximum deviation, ζ_i and ζ_i^* denote slack variables, i.e. ζ_i decides positive excess deviations while ζ_i^* decides the negative excess deviations. Subsequently, based on data subsets $\{D^1, D^2, D^3, \ldots, D^K\}$, K base learners $\{L^1, L^2, L^3, \ldots, L^K\}$ can be trained separately.

Finally, the aggregation strategy, mean rule, is used to aggregate the RUL prediction results of all base learners, which has been widely adopted in the regression ensemble [79]. Corresponding to input x_i, the final prediction value of RUL is:

$$\hat{\mathrm{y}}_i = \frac{1}{K}\sum_{k=1}^{K} f(\mathbf{x}_i^k) \tag{11}$$

The RUL prediction results of new machinery can be obtained by the trained ensemble model, which is helpful to production management and condition-based maintenance. The pseudo-code of the proposed FFEM is presented in Table 2.

Table 2
The pseudo-code of the FFEM.

Input: Training set $D=\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$;
Convex combination coefficient α;
SGL tuning parameter λ;
Random subspace rate $Ratio$;
The number of subsets K.
Process:
Base regressors $L \leftarrow \{\varnothing\}$;
Number of features in a feature subspace $num \leftarrow floor(m \times ratio)$;
for $k \in \{1,2,...,K\}$ do:
  Obtain ω from equation (7)
  for $j \in \{1,2,...,m\}$ do:
    Calculate weight w'_j by equation (8)
  end for
  Feature subspace $S^k \leftarrow \{\varnothing\}$;
  Repeat until $|S^k| = num$:
    Generate a random number γ in the range of 0 to 1;
    $sum \leftarrow 0$;
    for $j \in \{1,2,...,M\}$ do:
      $sum \leftarrow sum + w_j$;
      if $(sum + w'_j) \geq \gamma$ and the jth feature does not belong to S^k:
        Add the jth feature into S^k;
        $w'_j = \min(\mathbf{w}')$
      end if
    end for
  end of repeat
  Data subset D^k can be obtained according to S^k;
  Training base learner L^k on D^k by employing SVR;
  Add L^k to L;
end for
for $k \in \{1,2,...,K\}$ do:
  $f(\mathbf{x}^k) \leftarrow L^k(\mathbf{x})$; % Acquire prediction value of each base learner
end for
Output: The final prediction values of RUL are calculated by equation (11)

4. Experimental design

4.1. Experimental dataset

To verify our proposed method for RUL prediction, the data from the PRONOSTIA experimental platform at FEMTO-ST Institute in Besancon, France was used. By accelerating the degradation process of bearings throughout their whole life, the PRONOSTIA platform can provide real run-to-failure data that describe the degradation process of the bearings [31]. As shown in Fig. 5, there are three parts of the experimental platforms: the rotating part, the load part, and the measurement part. In the measurement part, the vibration signals can be collected from the two accelerometers, which are radially placed on the outer race of the bearing in the horizontal direction (the direction parallel to the platform) and vertical direction (the direction perpendicular to the platform). The sampling frequency is 25.6 kHz, that is, the vibration signals with 2560 numbers during 0.1s are collected every 10 s.

The bearing data contains 17 bearings under three operating conditions. Under the first condition, the rotation speed of the motor is 1800 r/min, and the load is 4000 N. Under the second condition, the motor speed and the load are 1650 r/min and 4200 N separately. Under the third condition, the motor speed is 1500 r/min, and the load is 5000 N.

Table 3
Detail of the datasets under condition 1.

Dataset	Number of instances	Total life
Bearing 1	2803	28 030 s
Bearing 2	871	8710 s
Bearing 3	2375	23 750 s
Bearing 4	1428	14 280 s
Bearing 5	2463	24 630 s
Bearing 6	2448	24 480 s
Bearing 7	2259	22 590 s

In this paper, all seven bearings under the first operating condition were used for experimenting validation, and the vibration signal from the horizontal direction was utilized for RUL prediction. The detailed information of the used dataset is illustrated in Table 3. The lengths of the total life of different bearings are different. There are 14 647 instances in total, and each instance includes 2560 data points. Besides, in each test, the leave-one-out strategy was adopted. That is, each bearing was utilized for testing in turn while the other six bearings were utilized for training. Specially, validation subset was split from training set for training BiLSTM models.

4.2. Evaluation metrics

To evaluate the performance of the proposed method and the benchmark methods quantitatively, the Root-Mean-Square Error (RMSE) and the Mean Absolute Error (MAE) were utilized as evaluation metrics. Their calculation formulas are as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(y_i - \hat{y}_i)^2} \tag{12}$$

$$\mathrm{MAE} = \frac{1}{L}\sum_{i=1}^{L}\left|y_i - \hat{y}_i\right| \tag{13}$$

where y_i and $\hat{y}_i$ denote the actual RUL and predicted RUL, separately, L denotes the total number of the testing data instances. Lower RMSE and MAE indicate better prediction performances.

4.3. Experimental procedure

As mentioned in Section 3.2, 16 time-domain features, 12 frequency-domain features, 32 time–frequency domain features, as well as 128 deep representation features are extracted from the original signals, which characterize the degradation information of the bearing from different perspectives. For instance, some features extracted from bearing 1 are shown in Fig. 6.

A variety of benchmark methods are utilized to prove the effectiveness of the proposed method. The benchmark methods include two main categories: (1) individual models, including Linear Regression (LR), Decision Tree (DT), and SVR; and (2) ensemble models, including Random Forest (RF), Bagging, AdaBoost, and RS. For a fair comparison, the base learners of all ensemble models excluding RF are SVR with the same parameters. The parameters utilized in the different prediction methods are listed in Table 4. Besides, to verify the complementarity of the deep representation features and signal analysis features, the RUL prediction performances of different feature groups are compared.

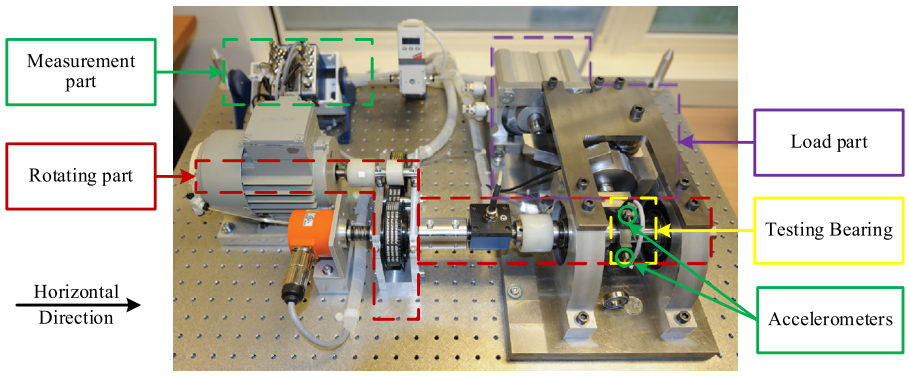

Fig. 5. PRONOSTIA experimental platform.

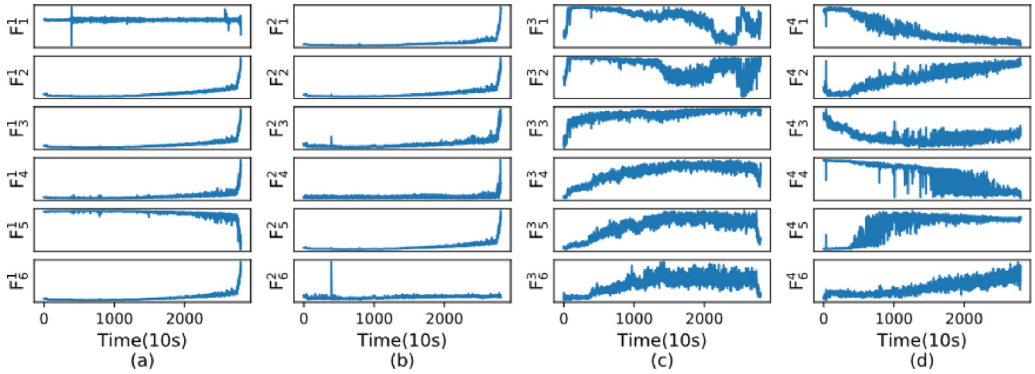

Fig. 6. Features of bearing 1 from the horizontal direction: (a) time-domain features, (b) frequency-domain features, (c) time–frequency domain features and (d) deep representation features.

Table 4
Details of the parameters used in the experiments.

Methods	Parameters
LR	Fit intercept: True;
DT	Criterion: MSE;
SVR	Kernel: RBF kernel; Penalty parameter: 1.0; Epsilon: 0.1
RF	Base learner: DT; Number of ensembles: 10; Criterion: MSE
Bagging	Base learner: SVR; Number of ensembles: 10
AdaBoost	Base learner: SVR; Number of ensembles: 10
RS	Base learner: SVR; Number of ensembles:10; Subspace rates: {0.1, 0.3, 0.5, 0.7, 0.9}
FFEM	Base learner: SVR; Number of ensembles: 10; Convex combination coefficients: {0.1, 0.3, 0.5, 0.7, 0.9}; SGL tuning parameters: {0.1, 0.3, 0.5, 0.7, 0.9}; Subspace rates: {0.1, 0.3, 0.5, 0.7, 0.9}

Table 5
RMSE of the comparison methods (%).

Bearing	LR	DT	SVR	RF	Bag	Ada	RS	FFEM
1	40.13	17.79	11.09	14.09	7.05	9.15	6.69	**6.32**
2	26.73	26.08	16.53	17.79	16.53	16.26	17.17	**15.98**
3	20.66	16.42	16.17	9.91	8.23	9.22	8.32	**8.21**
4	34.02	26.33	17.87	11.48	9.90	18.85	9.08	**7.62**
5	23.44	19.95	18.08	16.67	14.39	15.32	14.32	**13.85**
6	22.48	24.21	22.92	19.00	17.14	17.65	17.13	**17.10**
7	32.90	25.81	24.88	17.57	17.21	22.17	17.24	**16.98**

5. Results and discussions

5.1. Experimental results

These experiments are performed using Python 3.6 under 64-bit Windows 10 Professional operating system by a desktop computer: 16-core Genuine Intel(R) i7 CPU at 2.4 GHz and 64 GB RAM main memory. Each model is repeatedly validated 10 times to reduce the impact of randomness, and the mean values of evaluation metrics are computed for the final evaluation.

The comparison of RMSE and MAE between the proposed FFEM and the compared methods is shown in Tables 5 and 6, in which the best values on each test bearing are highlighted in boldface. It is concluded that the proposed FFEM achieved the lowest regression error compared to almost all other benchmarked methods. In detail, the RMSE and the MAE of the FFEM are 6.32% and 4.84% (bearing 1), 15.98% and 13.07% (bearing 2), 8.21% and 6.34% (bearing 3), 7.62% and 5.84% (bearing 4), 13.85% and 9.07% (bearing 5), 17.10% and 13.99% (bearing 6), and 16.98% and 11.52% (bearing 7). The average RMSE and MAE of all bearings are 12.29% and 9.24%. Taking bearing 1 for example, the decline rate of RMSE and MAE achieved by FFEM over RF, Bagging, AdaBoost, and RS with fusion features are respectively 55% and 58%, 10% and 14%, 31% and 35%, and 5% and 7%. Besides, it can be observed that the regression error of the SVR model is the lowest compared with other single models including LR and DT. As for ensemble models, the RF performs worse than other ensemble methods generally. Among Bagging, AdaBoost, RS, and the proposed FFEM, the AdaBoost method achieves the poorest performances in most cases, this phenomenon may be caused by the overfitting problem owing to noise instances during the training process of AdaBoost. In conclusion, the proposed method FFEM can predict the RUL of the tested bearings accurately.

Besides, Fig. 7 presents the visualization of the RUL prediction results for all seven bearings by the proposed FFEM. The red point denotes the predicted RUL while the true RUL is represented by the green point. Under the horizontal axis, the blue histogram denotes the absolute error at each time point. It can be observed

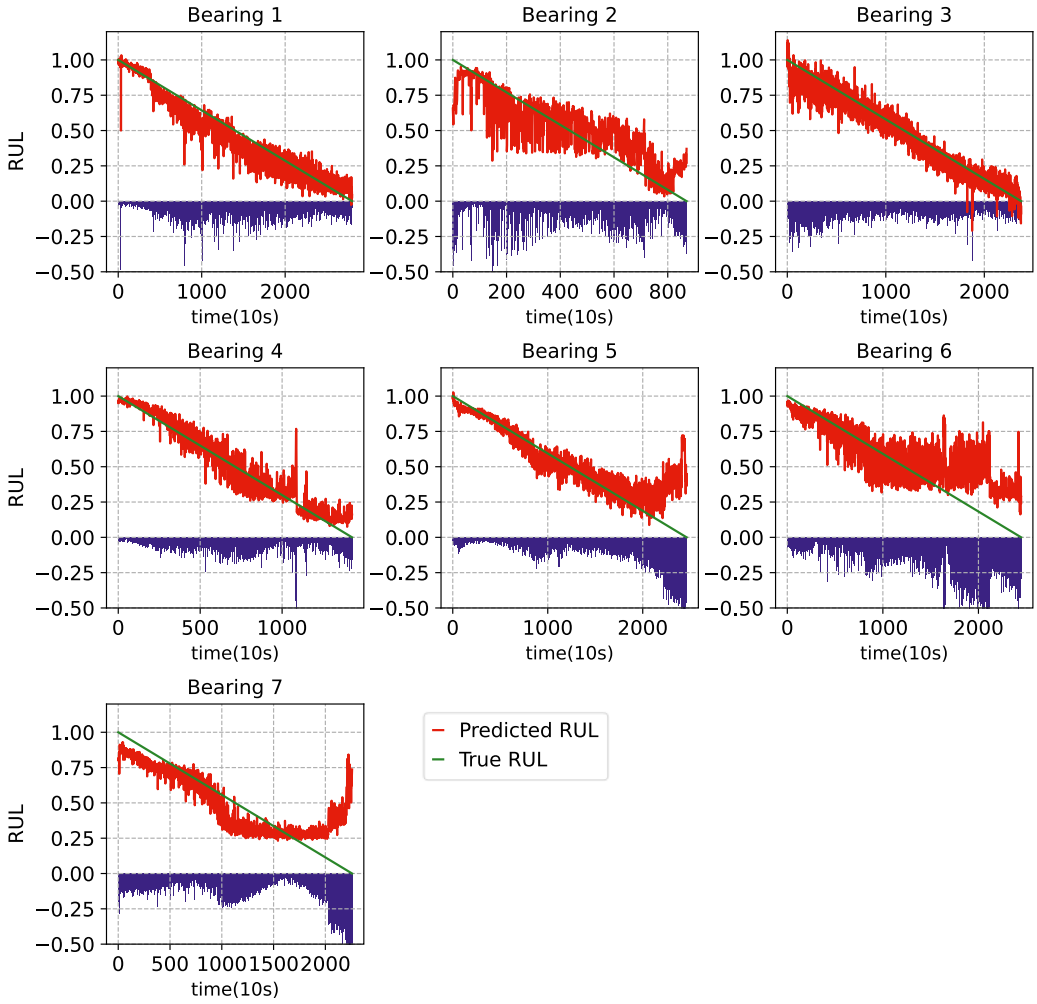

Fig. 7. RUL prediction results of test bearings by the proposed method.

Table 6
MAE of the comparison methods (%).

Bearing	LR	DT	SVR	RF	Bag	Ada	RS	FFEM
1	36.13	13.76	8.49	11.45	5.62	7.44	5.22	**4.84**
2	22.83	17.78	13.59	12.99	13.60	**12.97**	14.27	13.07
3	16.87	12.29	14.30	7.71	6.36	6.84	**6.33**	6.34
4	30.45	19.83	15.55	9.37	7.96	15.19	7.23	**5.84**
5	18.92	13.36	11.95	10.52	9.36	9.60	9.24	**9.07**
6	17.00	18.61	17.59	15.85	14.16	15.15	14.10	**13.99**
7	29.53	19.05	22.17	14.00	12.13	17.26	12.19	**11.52**

that the general degradation evolution is effectively captured, and the proposed method can accurately predict the RUL for the tested bearings.

5.2. Discussions

5.2.1. Comparison of features for RUL prediction

To verify the reasonability for the fusion of signal analysis features and deep representation features, the prediction errors of a variety of features were compared. As shown in Fig. 8, the correlation analysis was conducted on time-domain features (F1), frequency-domain features (F2), time–frequency domain features (F3), and deep representation features extracted by BiLSTM (F4). It is noteworthy that there is some redundancy in features, especially for signal analysis features. Figs. 9 and 10 illustrate the RMSE and MAE of seven bearings on various features, including the aforementioned F1, F2, F3, all signal analysis features (F1+F2+F3), F4, and their fusion (F1+F2+F3+F4). As for features in a single group, the prediction errors of deep representation features extracted from BiLSTM are the lowest generally, and the prediction performance of time–frequency domain features surpassed those of time-domain features and frequency-domain features in most cases. Moreover, it appears that models using all signal analysis feature gained better performances than only using F1, F2, or F3. For different bearings, single models achieved the best performance on different features, which indicates that those features are useful for RUL prediction. As for ensemble models, they almost achieved the best performance on fusion features, demonstrating that multiple features may complement each other. Taking bearing 1 for example, the decreased RMSE and MAE of the fusion features under FFEM are respectively 46% and 51% over F1, 53% and 56% over F2, 50% and 53% over F3, 39% and 43% over F1+F2+F3, and 3% and 3% over F4. Besides, for bearing 7, poor deep representation features may be noisy for prediction results, but the FFEM can reduce this negative impact better than other methods. Hence, it is concluded that there is a positive impact on RUL prediction by fusing signal analysis features and deep representation features in most cases.

5.2.2. Parameters analysis

Three key parameters, convex combination coefficient α (Alpha), SGL tuning parameter λ (Lambda), and random subspace

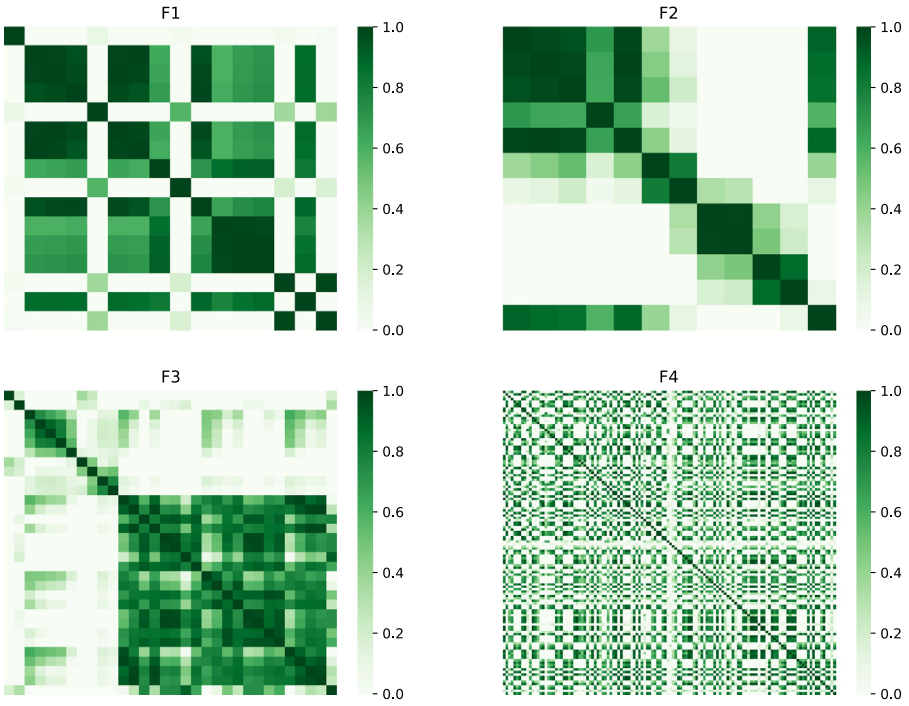

Fig. 8. Feature correlation analysis upon different features of bearing 1.

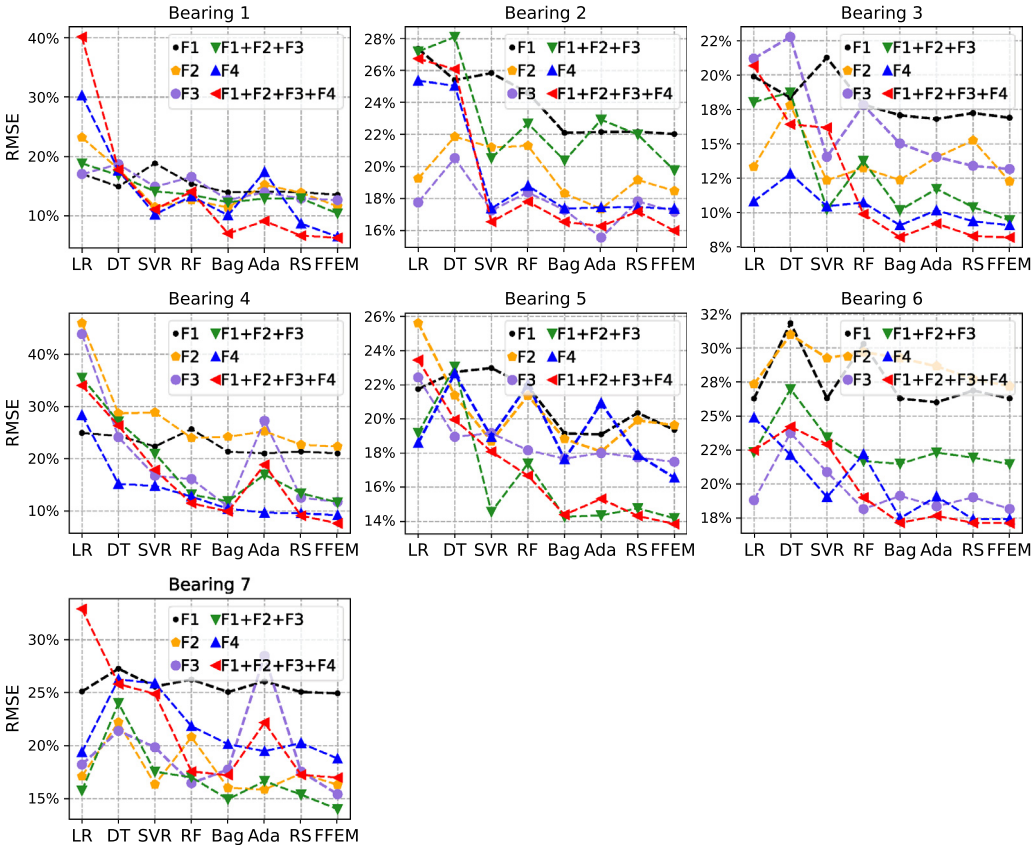

Fig. 9. RMSE comparison upon different features.

rate *Ratio*, are important for the proposed FFEM method. Figs. 11 and 12 show the joint effect of the *Ratio* and α to the FFEM when the λ has been optimized, in which the X-axis is the random subspace rate *Ratio*, the Y-axis denotes convex combination coefficient α and the Z-axis denotes the RMSE or MAE. In the experiments, the value of the convex combination coefficient α and random subspace rate *Ratio* were both from 0.1 to 0.9. With respect to different test bearings, the proposed FFEM reached the

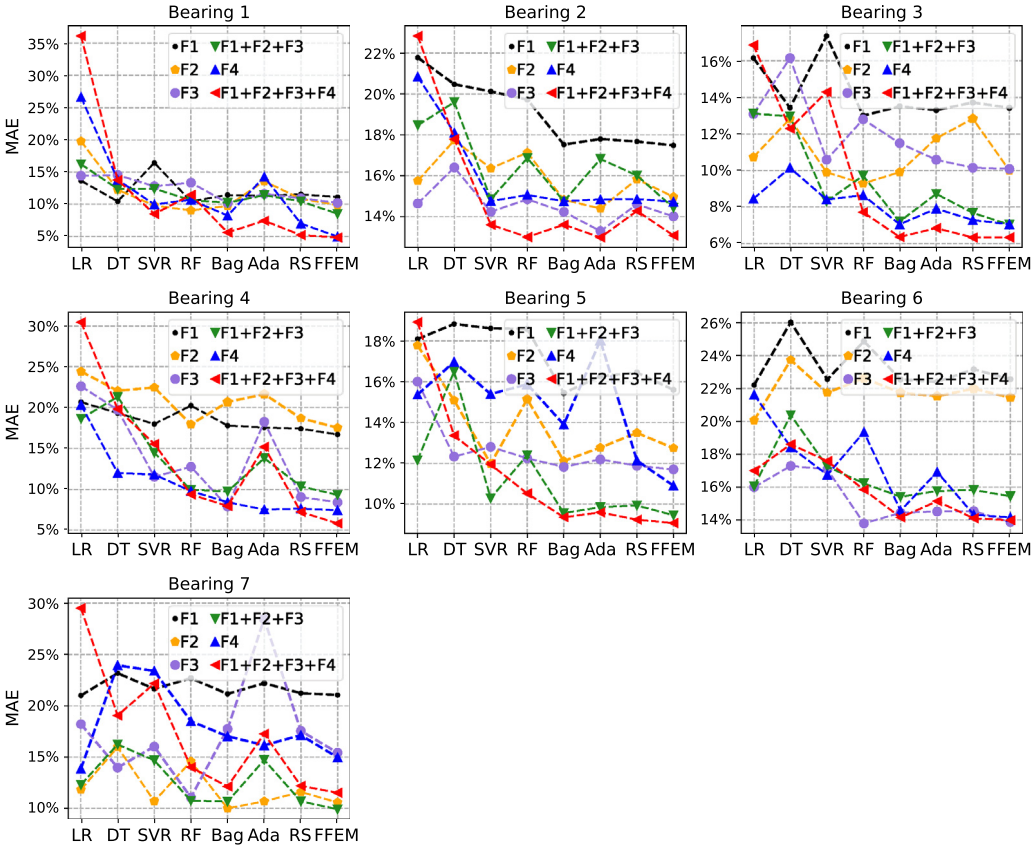

Fig. 10. MAE comparison upon different features.

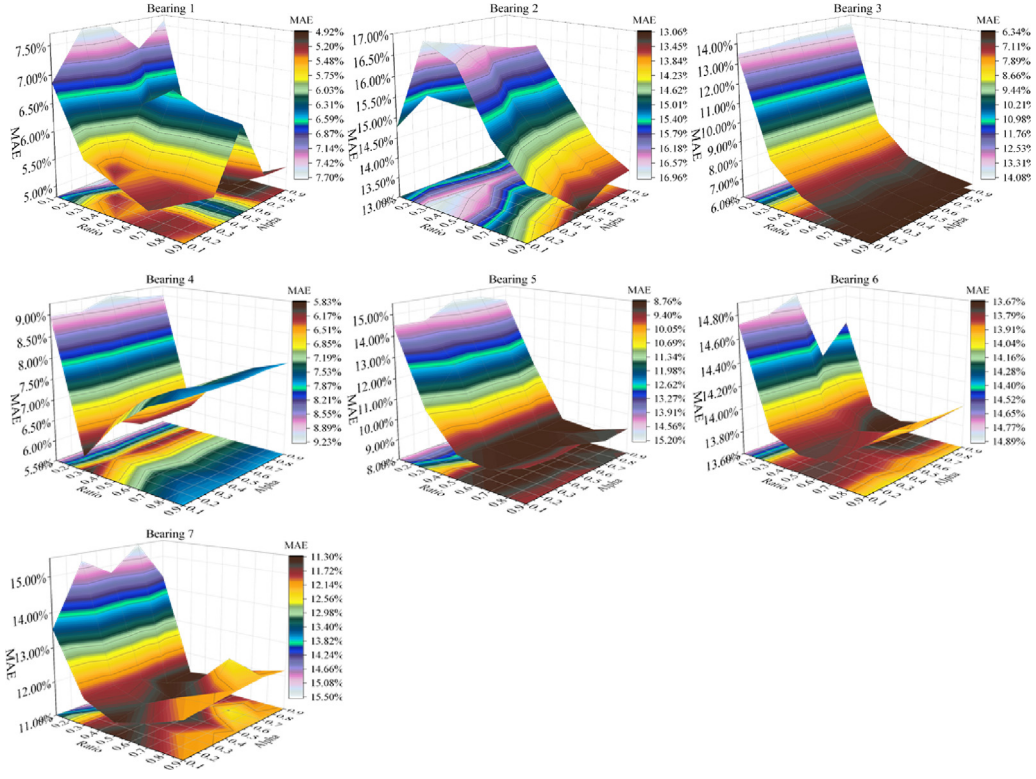

Fig. 11. Sensitivity analysis of RMSE for FFEM on α and *Ratio*.

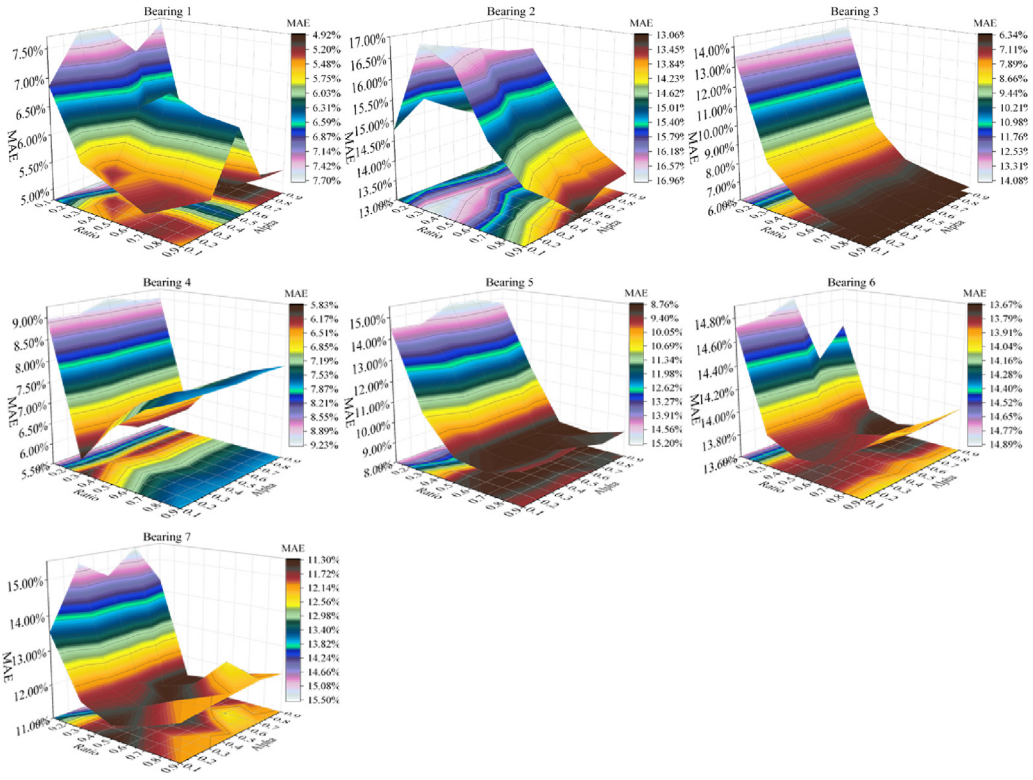

Fig. 12. Sensitivity analysis of MAE for FFEM on α and *Ratio*.

lowest RMSE with $Ratio = 0.7/\alpha = 0.7$ on bearing 1, $Ratio = 0.9/\alpha = 0.5$ on bearing 2, $Ratio = 0.9/\alpha = 0.5$ on bearing 3, $Ratio = 0.3/\alpha = 0.1$ on bearing 4, $Ratio = 0.7/\alpha = 0.9$ on bearing 5, $Ratio = 0.9/\alpha = 0.7$ on bearing 6, and $Ratio = 0.5/\alpha = 0.7$ on bearing 7. From the *Y*-axis in Figs. 11 and 12, it appears that the RMSE and MAE decreased gradually with the change of *Ratio*, meaning that the performance of FFEM is influenced by it. From the *X*-axis in Figs. 11 and 12, the convex combination coefficient α also has an important effect on the performance of FFEM. The reason for this phenomenon is that the $1-\alpha$ influences sparsity upon feature groups and α determines sparse effects for individual features within a group. Besides, the influence of α and λ on the FFEM is shown in Figs. 13 and 14, in which the *X*-axis denotes the α and *Y*-axis denotes the λ. It can be observed that the optimal values of the prediction error can be gained when $\alpha \geq 0.5$ for most bearings, and it is not trivial to set the optimal values of these parameters since optimal values are not the same for different bearings. Overall, these key parameters greatly contribute to the development of model performance. It can be concluded that the performance of the FFEM for RUL prediction is significantly improved if the related parameters can be attuned properly.

5.2.3. Training time and testing time

In this subsection, we analyzed the training time and testing time of the proposed FFEM. Before the prediction model construction, feature extraction is a significant step. For signal analysis features, the time taken to extract time-domain features, frequency-domain features and time–frequency domain features for seven bearings are 12.19 s, 19.97 s, and 16.23 s, respectively. For deep representation features, the features of testing data can be extracted instantaneously by a trained deep learning network. In the experiment, the average training time of BiLSTM is 6.53 min, and the average time of extracting deep representation features for a testing bearing is 0.03 s. Besides, the information about training time and testing time of different prediction models is listed in Table 7. As shown in Table 7, the ensemble models consume more training time and testing time than the single models. For the ensemble models, the proposed FFEM consumes more training time than RF and Bagging, but much less time than AdaBoost. In terms of testing time, there is no significant difference between the Bagging, RS, and the proposed FFEM. However, as mentioned in Section 5.1, the FFEM achieves lower prediction error than almost all other methods at an acceptable cost of testing time. In practice, machine learning-based RUL prediction methods are usually trained offline and then deployed for testing. Therefore, the proposed FFEM can be applied in practice to predict the RUL of machinery.

6. Conclusions and future work

In this paper, a novel RUL prediction method, FFEM was proposed to fuse the signal analysis features and deep representation features for improving the RUL prediction performance. In this method, diverse features could be fused properly by introducing group-based sparse learning to use the complementarity among them. In the meanwhile, the negative impact of irrelevant and redundant features was reduced, contributing on acquiring more accurate and diverse base learners. Besides, experiments on the run-to-failure datasets of bearings were conducted to validate the effectiveness and superiority of FFEM compared to other commonly used machine learning methods.

Although the proposed FFEM has been proven to be effective for RUL prediction in our empirical experiments, there are several future research directions for this study. First, although the signal analysis features and deep representation features are fused for the RUL prediction of machinery in this paper, poor deep representation features may be noisy for predictions, and other information such as the working conditions of machinery can also be utilized to complement more information. Second,

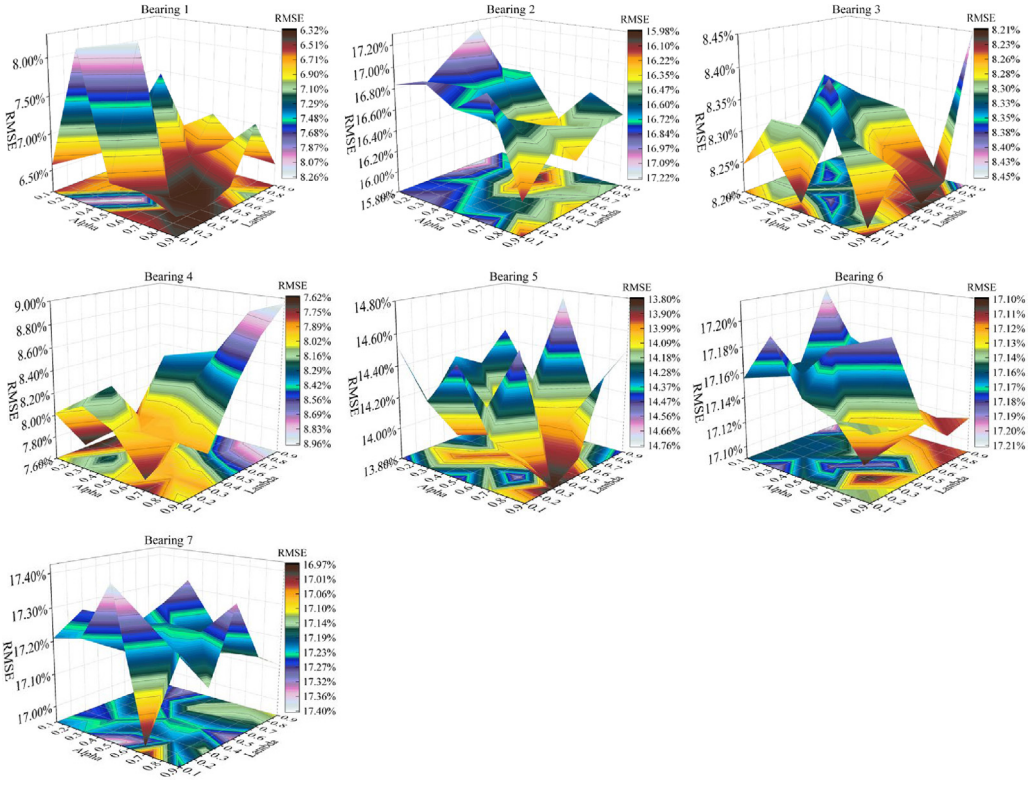

Fig. 13. Sensitivity analysis of RMSE for FFEM on α and λ.

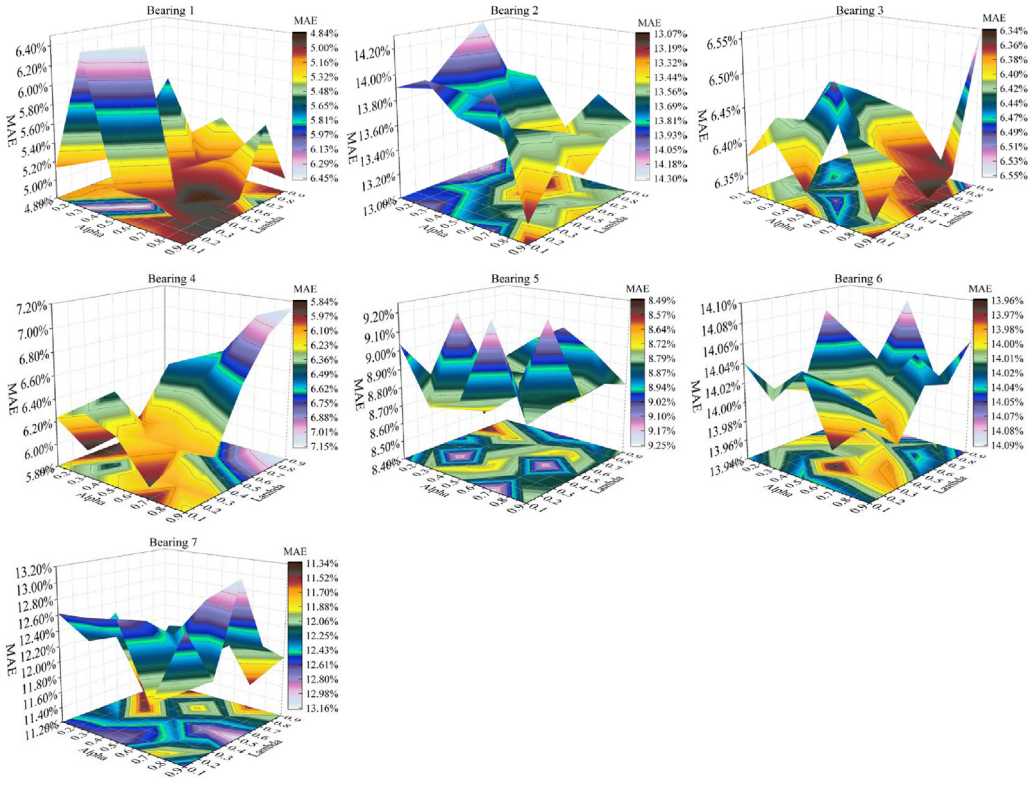

Fig. 14. Sensitivity analysis of MAE for FFEM on α and λ.

Table 7
The training/testing time of the comparison methods (s: second).

Bearing	LR	DT	SVR	RF	Bag	Ada	RS	FFEM
1	[a]/[a]	2.64/[a]	5.14/1.93	0.86/[a]	22.71/12.57	208.06/50.75	43.67/13.86	42.21/12.21
2	0.18/[a]	3.00/[a]	5.73/0.55	8.27/[a]	24.73/3.45	279.90/14.29	47.05/3.98	45.19/3.49
3	[a]/[a]	2.82/[a]	5.18/1.64	0.18/[a]	22.45/10.18	232.80/46.40	43.11/11.20	42.04/9.97
4	0.09/[a]	3.00/[a]	5.73/0.82	[a]/[a]	24.45/6.00	256.26/29.26	46.38/6.69	44.39/5.88
5	[a]/[a]	2.64/[a]	4.27/1.36	11.45/[a]	18.64/8.91	199.20/43.64	37.96/10.07	36.82/8.90
6	[a]/[a]	3.00/[a]	3.82/1.09	0.91/[a]	16.55/7.82	219.38/49.56	32.64/8.73	32.59/7.71
7	[a]/[a]	2.73/[a]	4.64/1.09	11.55/[a]	19.18/7.91	205.66/34.71	37.33/9.04	35.40/7.61

Notes:
[a]Represents the time is shorter than 0.01 s (<0.01 s).

more datasets should be used to verify the framework proposed in this paper to support the effectiveness of FFEM. Third, parallel computing technology should be considered to settle computationally intensive problems and the difference in data distribution can be considered to improve the accuracy in future work.

CRediT authorship contribution statement

Gang Wang: Conceptualization, Methodology. **Hui Li:** Methodology, Software, Formal analysis. **Feng Zhang:** Methodology, Visualization. **Zhangjun Wu:** Methodology.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (72071062), Science Fund for Distinguished Young Scholars of AnHui (2208085J12), Anhui Provincial Key Research and Development Program (202104a05020038), and Fundamental Research Funds for the Central Universities, China (PA2021KCPY0032).

References

[1] K. Peng, R. Jiao, J. Dong, Y. Pi, A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter, Neurocomputing 361 (2019) 19–28, http://dx.doi.org/10.1016/j.neucom.2019.07.075.
[2] G.W. Vogl, B.A. Weiss, M. Helu, A review of diagnostic and prognostic capabilities and best practices for manufacturing, J. Intell. Manuf. 30 (1) (2019) 79–95, http://dx.doi.org/10.1007/s10845-016-1228-8.
[3] T. Xia, Y. Dong, L. Xiao, S. Du, E. Pan, L. Xi, Recent advances in prognostics and health management for advanced manufacturing paradigms, Reliab. Eng. Syst. Saf. 178 (2017) (2018) 255–268, http://dx.doi.org/10.1016/j.ress.2018.06.021.
[4] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, D. Siegel, Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications, Mech. Syst. Signal Process. 42 (1–2) (2014) 314–334, http://dx.doi.org/10.1016/j.ymssp.2013.06.004.
[5] S. Uckun, K. Goebel, P.J.F. Lucas, Standardizing research methods for prognostics, in: 2008 Int. Conf. Progn. Heal. Manag. PHM 2008, 2008, http://dx.doi.org/10.1109/PHM.2008.4711437.
[6] 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.
[7] A. Cubillo, S. Perinpanayagam, M. Esperon-miguez, A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery, Adv. Mech. Eng. 8 (8) (2016) 1–21, http://dx.doi.org/10.1177/1687814016664660.
[8] D. Wang, K.L. Tsui, Statistical modeling of bearing degradation signals, IEEE Trans. Reliab. 66 (4) (2017) 1331–1344, http://dx.doi.org/10.1109/TR.2017.2739126.
[9] K. Javed, R. Gouriveau, N. Zerhouni, State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels, Mech. Syst. Signal Process. 94 (2017) 214–236, http://dx.doi.org/10.1016/j.ymssp.2017.01.050.
[10] 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, http://dx.doi.org/10.1016/j.ymssp.2017.11.016.
[11] J. Wu, C. Wu, S. Cao, S.W. Or, C. Deng, X. Shao, Degradation data-driven time-to-failure prognostics approach for rolling element bearings in electrical machines, IEEE Trans. Ind. Electron. 66 (1) (2019) 529–539, http://dx.doi.org/10.1109/TIE.2018.2811366.
[12] A.K.S. 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.
[13] 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, http://dx.doi.org/10.1016/j.jmsy.2017.02.013.
[14] E. Sutrisno, H. Oh, A.S.S. Vasan, M. Pecht, Estimation of remaining useful life of ball bearings using data driven methodologies, in: PHM 2012-2012 IEEE Int. Conf.on Progn. Heal. Manag. Enhancing Safety, Effic. Availability, Eff. Syst. Through PHM Technol. Appl. Conf. Progr. Vol. 2, 2012, pp. 1–7, http://dx.doi.org/10.1109/ICPHM.2012.6299548.
[15] M. Zhao, B. Tang, Q. Tan, Bearing remaining useful life estimation based on time-frequency representation and supervised dimensionality reduction, Meas. J. Int. Meas. Confed. 86 (2016) 41–55, http://dx.doi.org/10.1016/j.measurement.2015.11.047.
[16] 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.
[17] G. Hinton others, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97, http://dx.doi.org/10.1109/MSP.2012.2205597.
[18] Y. Wu, M. Yuan, S. Dong, L. Lin, Y. Liu, Remaining useful life estimation of engineered systems using vanilla LSTM neural networks, Neurocomputing 275 (2018) 167–179, http://dx.doi.org/10.1016/j.neucom.2017.05.063.
[19] B. Yang, R. Liu, E. Zio, Remaining useful life prediction based on a double-convolutional neural network architecture, IEEE Trans. Ind. Electron. 66 (12) (2019) 9521–9530, http://dx.doi.org/10.1109/TIE.2019.2924605.
[20] Q. An, Z. Tao, X. Xu, M. El Mansori, M. Chen, A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network, Meas. J. Int. Meas. Confed. 154 (2020) 107461, http://dx.doi.org/10.1016/j.measurement.2019.107461.
[21] W. Zhang, X. Li, X. Li, Deep learning-based prognostic approach for lithium-ion batteries with adaptive time-series prediction and on-line validation, Meas. J. Int. Meas. Confed. 164 (2020) http://dx.doi.org/10.1016/j.measurement.2020.108052.
[22] F. Lu, J. Wu, J. Huang, X. Qiu, Aircraft engine degradation prognostics based on logistic regression and novel OS-ELM algorithm, Aerosp. Sci. Technol. 84 (2019) 661–671, http://dx.doi.org/10.1016/j.ast.2018.09.044.
[23] M. Elforjani, S. Shanbr, Prognosis of bearing acoustic emission signals using supervised machine learning, IEEE Trans. Ind. Electron. 65 (7) (2018) 5864–5871, http://dx.doi.org/10.1109/TIE.2017.2767551.
[24] 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, http://dx.doi.org/10.1109/TR.2013.2285318.
[25] G. Qiu, Y. Gu, J. Chen, Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and Weibull proportional hazards model, Meas. J. Int. Meas. Confed. 150 (2020) 107097, http://dx.doi.org/10.1016/j.measurement.2019.107097.

[26] P. Baraldi, F. Mangili, E. Zio, A Kalman filter-based ensemble approach with application to turbine creep prognostics, IEEE Trans. Reliab. 61 (4) (2012) 966–977, http://dx.doi.org/10.1109/TR.2012.2221037.
[27] Y. Zhang, D. Liu, J. Yu, Y. Peng, X. Peng, Microelectronics reliability EMA remaining useful life prediction with weighted bagging GPR algorithm, Microelectron. Reliab. 75 (2017) 253–263, http://dx.doi.org/10.1016/j.microrel.2017.03.021.
[28] Z. Li, D. Wu, C. Hu, J. Terpenny, An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction, Reliab. Eng. Syst. Saf. 184 (2017) (2019) 110–122, http://dx.doi.org/10.1016/j.ress.2017.12.016.
[29] Y. Ren, L. Zhang, P.N. Suganthan, Ensemble classification and regression-recent developments, applications and future directions [review article], IEEE Comput. Intell. Mag. 11 (1) (2016) 41–53, http://dx.doi.org/10.1109/MCI.2015.2471235.
[30] G. Wang, F. Zhang, B. Cheng, F. Fang, DAMER: a novel diagnosis aggregation method with evidential reasoning rule for bearing fault diagnosis, J. Intell. Manuf. 32 (1) (2020) 1–20, http://dx.doi.org/10.1007/s10845-020-01554-5.
[31] P. Nectoux others, PRONOSTIA: An experimental platform for bearings accelerated degradation tests. To cite this version: HAL id: hal-00719503 PRONOSTIA: An experimental platform for bearings accelerated degradation tests, in: IEEE Int. Conf. Progn. Heal. Manag. PHM'12. 2012, Denver, Col- orado, United States, 2012, pp. 1–8.
[32] T. Benkedjouh, K. Medjaher, N. Zerhouni, S. Rechak, Remaining useful life estimation based on nonlinear feature reduction and support vector regression, Eng. Appl. Artif. Intell. 26 (7) (2013) 1751–1760, http://dx.doi.org/10.1016/j.engappai.2013.02.006.
[33] R.K. Singleton, E.G. Strangas, S. Aviyente, The use of bearing currents and vibrations in lifetime estimation of bearings, IEEE Trans. Ind. Inform. 13 (3) (2017) 1301–1309, http://dx.doi.org/10.1109/TII.2016.2643693.
[34] V. Atamuradov, K. Medjaher, F. Camci, P. Dersin, N. Zerhouni, Railway point machine prognostics based on feature fusion and health state assessment, IEEE Trans. Instrum. Meas. 68 (8) (2019) 2691–2704, http://dx.doi.org/10.1109/TIM.2018.2869193.
[35] M. Xia, T. Li, T. Shu, J. Wan, C.W. De Silva, Z. Wang, A two-stage approach for the remaining useful life prediction of bearings using deep neural networks, IEEE Trans. Ind. Inform. 15 (6) (2019) 3703–3711, http://dx.doi.org/10.1109/TII.2018.2868687.
[36] X. Li, Q. Ding, J.Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliab. Eng. Syst. Saf. 172 (2017) (2018) 1–11, http://dx.doi.org/10.1016/j.ress.2017.11.021.
[37] J. Ma, H. Su, W.L. Zhao, B. Liu, Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning, Complexity 2018 (2018) http://dx.doi.org/10.1155/2018/3813029.
[38] J. Deutsch, D. He, Using deep learning-based approach to predict remaining useful life of rotating components, IEEE Trans. Syst. Man Cybern. Syst. 48 (1) (2017) 11–20, http://dx.doi.org/10.1109/TSMC.2017.2697842.
[39] W. Yu, I.Y. Kim, C. Mechefske, Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme, Mech. Syst. Signal Process. 129 (2019) 764–780, http://dx.doi.org/10.1016/j.ymssp.2019.05.005.
[40] A. Al-Dulaimi, S. Zabihi, A. Asif, A. Mohammadi, A multimodal and hybrid deep neural network model for remaining useful life estimation, Comput. Ind. 108 (2019) 186–196, http://dx.doi.org/10.1016/j.compind.2019.02.004.
[41] J. Wang, R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, Machine health monitoring using local feature-based gated recurrent unit networks, IEEE Trans. Ind. Electron. 65 (2) (2017) 1539–1548, http://dx.doi.org/10.1109/TIE.2017.2733438.
[42] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, K. Dietmayer, Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods, J. Power Sources 239 (2013) 680–688, http://dx.doi.org/10.1016/j.jpowsour.2012.11.146.
[43] H.Z. Huang, H.K. Wang, Y.F. Li, L. Zhang, Z. Liu, Support vector machine based estimation of remaining useful life: current research status and future trends, J. Mech. Sci. Technol. 29 (1) (2015) 151–163, http://dx.doi.org/10.1007/s12206-014-1222-z.
[44] C. Zhang, P. Lim, A.K. Qin, K.C. Tan, Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics, IEEE Trans. Neural Networks Learn. Syst. 28 (10) (2017) 2306–2318, http://dx.doi.org/10.1109/TNNLS.2016.2582798.
[45] K. Javed, R. Gouriveau, N. Zerhouni, P. Nectoux, Enabling health monitoring approach based on vibration data for accurate prognostics, IEEE Trans. Ind. Electron. 62 (1) (2015) 647–656, http://dx.doi.org/10.1109/TIE.2014.2327917.
[46] R. Huang, L. Xi, X. Li, C. Richard 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, http://dx.doi.org/10.1016/j.ymssp.2005.11.008.
[47] 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) (2017) 2276–2285, http://dx.doi.org/10.1109/TIE.2016.2623260.
[48] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, S. Poll, Ensemble of optimized echo state networks for remaining useful life prediction, Neurocomputing 281 (2018) 121–138, http://dx.doi.org/10.1016/j.neucom.2017.11.062.
[49] J. Sun, H. Zuo, H. Yang, P. Michael, Study of ensemble learning-based fusion prognostics, in: 2010 Progn. Syst. Heal. Manag. Conf. PHM '10, 2010, pp. 1–7, http://dx.doi.org/10.1109/PHM.2010.5414582.
[50] D. Wu, C. Jennings, J. Terpenny, S. Kumara, Cloud-based machine learning for predictive analytics: Tool wear prediction in milling, in: Proc. - 2016 IEEE Int. Conf. Big Data, Big Data 2016, 2016, pp. 2062–2069, http://dx.doi.org/10.1109/BigData.2016.7840831.
[51] F. Wang, B. Wang, B. Dun, X. Chen, D. Yan, H. Zhu, Remaining life prediction of rolling bearing based on PCA and improved logistic regression model, J. Vibroeng. 18 (8) (2016) 5192–5203, http://dx.doi.org/10.21595/jve.2016.17449.
[52] R.K. Singleton, E.G. Strangas, S. Aviyente, Extended kalman filtering for remaining-useful-life estimation of bearings, IEEE Trans. Ind. Electron. 62 (3) (2015) 1781–1790, http://dx.doi.org/10.1109/TIE.2014.2336616.
[53] F. Yang, M.S. Habibullah, T. Zhang, Z. Xu, P. Lim, S. Nadarajan, Health index-based prognostics for remaining useful life predictions in electrical machines, IEEE Trans. Ind. Electron. 63 (4) (2016) 2633–2644, http://dx.doi.org/10.1109/TIE.2016.2515054.
[54] X. Li, F. Zhang, G. Wang, F. Fang, Joint optimization of statistical and deep representation features for bearing fault diagnosis based on random, Meas. Sci. Technol. 32 (2) (2021) 025115.
[55] Y. Lei, J. Lin, Z. He, M.J. Zuo, A review on empirical mode decomposition in fault diagnosis of rotating machinery, Mech. Syst. Signal Process. 35 (1–2) (2013) 108–126, http://dx.doi.org/10.1016/j.ymssp.2012.09.015.
[56] F.K. Wang, T. Mamo, Hybrid approach for remaining useful life prediction of ball bearings, Qual. Reliab. Eng. Int. 35 (7) (2019) 2494–2505, http://dx.doi.org/10.1002/qre.2538.
[57] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R.X. Gao, Deep learning and its applications to machine health monitoring, Mech. Syst. Signal Process. 115 (2019) 213–237, http://dx.doi.org/10.1016/j.ymssp.2018.05.050.
[58] R.X. Gao, R. Yan, Non-stationary signal processing for bearing health monitoring, Int. J. Manuf. Res. 1 (1) (2006) 18–40, http://dx.doi.org/10.1504/IJMR.2006.010701.
[59] A. Rohani Bastami, A. Aasi, H.A. Arghand, Estimation of remaining useful life of rolling element bearings using wavelet packet decomposition and artificial neural network, Iran. J. Sci. Technol. - Trans. Electr. Eng. 43 (s1) (2019) 233–245, http://dx.doi.org/10.1007/s40998-018-0108-y.
[60] A. Soualhi, K. Medjaher, N. Zerhouni, Bearing health monitoring based on hilbert-huang transform, support vector machine, and regression, IEEE Trans. Instrum. Meas. 64 (1) (2015) 52–62, http://dx.doi.org/10.1109/TIM.2014.2330494.
[61] 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) http://dx.doi.org/10.1155/2016/4632562.
[62] R. Yan, R.X. Gao, X. Chen, Wavelets for fault diagnosis of rotary machines: A review with applications, Signal Process. 96 (PART A) (2014) 1–15, http://dx.doi.org/10.1016/j.sigpro.2013.04.015.
[63] A. Elsheikh, S. Yacout, M.S. Ouali, Bidirectional handshaking LSTM for remaining useful life prediction, Neurocomputing 323 (2019) http://dx.doi.org/10.1016/j.neucom.2018.09.076.
[64] S. Hochreiter, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[65] C.G. Huang, H.Z. Huang, Y.F. Li, A bidirectional LSTM prognostics method under multiple operational conditions, IEEE Trans. Ind. Electron. 66 (11) (2019) 8792–8802, http://dx.doi.org/10.1109/TIE.2019.2891463.
[66] J. Wu, K. Hu, Y. Cheng, H. Zhu, X. Shao, Y. Wang, Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network, ISA Trans. 97 (2020) 241–250, http://dx.doi.org/10.1016/j.isatra.2019.07.004.
[67] J. Werbos Paul, Backpropagation through time: what it does and how it does, Proc. IEEE 78 (10) (1990) 1550–1561.
[68] G. Wang, G. Chen, Y. Chu, A new random subspace method incorporating sentiment and textual information for financial distress prediction, Electron. Commer. Res. Appl. 29 (2018) 30–49, http://dx.doi.org/10.1016/j.elerap.2018.03.004.
[69] I. Decision-making, T. Road, T.C. Avenue, H. Kong, Igf-bagging: information gain based feature selection for bagging, Int. J. Innov. Comput. Inf. Control 7 (11) (2011) 6247–6259.
[70] T.G. Dietterich, Ensemble methods in machine learning, Int. Work. Mult. Classif. Syst. 1857 (2000) 1–15.
[71] G. Wang, J. Hao, J. Ma, H. Jiang, A comparative assessment of ensemble learning for credit scoring, Expert Syst. Appl. 38 (1) (2011) 223–230, http://dx.doi.org/10.1016/j.eswa.2010.06.048.

[72] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[73] R. Tibshirani, The lasso method for variable selection in the cox model, Stat. Med. 16 (4) (1997) 385–395, http://dx.doi.org/10.1002/(SICI)1097-0258(19970228)16:4<385::AID-SIM380>3.0.CO;2-3.
[74] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. Ser. B Stat. Methodol. 68 (1) (2006) 49–67, http://dx.doi.org/10.1111/j.1467-9868.2005.00532.x.
[75] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, A sparse-group lasso, J. Comput. Graph. Stat. 22 (2) (2013) 231–245, http://dx.doi.org/10.1080/10618600.2012.681250.
[76] F.J. Detmer, J. Cebral, M. Slawski, A note on coding and standardization of categorical variables in (sparse) group lasso regression, J. Statist. Plann. Inference 206 (1) (2020) 1–11, http://dx.doi.org/10.1016/j.jspi.2019.08.003.
[77] L. Cao, Support vector machines experts for time series forecasting, Neurocomputing 51 (2003) 321–339, http://dx.doi.org/10.1016/S0925-2312(02)00577-5.
[78] C.C. Chang, C.J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–39, http://dx.doi.org/10.1145/1961189.1961199.
[79] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–44, http://dx.doi.org/10.1109/MCAS.2006.1688199.