Applied Soft Computing 127 (2022) 109335

Contents lists available at ScienceDirect

Applied Soft Computing

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

Rotating machinery faults detection method based on deep echo state network

Xin Li [a], Fengrong Bi [a], Lipeng Zhang [b], Jiewei Lin [a,*], Xiaobo Bi [c], Xiao Yang [a]

[a] *State Key Laboratory of Engines, Tianjin University, Tianjin 300350, China*
[b] *Tianjin Internal Combustion Engine Research Institute, Tianjin 300072, China*
[c] *School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China*

ARTICLE INFO

Article history:
Received 11 August 2021
Received in revised form 3 March 2022
Accepted 15 July 2022
Available online 20 July 2022

Keywords:
Echo state network (ESN)
Deep learning
Fault detection
Pattern recognition
Rotating machinery

ABSTRACT

This paper aims to develop an accurate and efficient end-to-end fault detection model trained by small-scale data for the rotating machinery. The echo state network (ESN) is promising thanks to the training process by linear regression, but it struggles in mining spatial information. Thus, a deep ESN based on fixed convolution kernels (FCK-DESN) is proposed. The Prewitt, the Sobel, and the Gaussian lowpass filters are designed as the fixed convolution kernels for spatial feature extraction without training. The one hidden layer autoencoder is built to compress the dimensionality and improve the applicability. Based on the pre-process modules, the ESN could realize pattern recognition under complex conditions. The fault detection approach is then constructed based on the time–frequency information provided by the smoothed pseudo-Wigner–Ville distribution. Case studies of a rotor-bearing system and a diesel engine show that the proposed FCK-DESN approach has better recognition rates than popular deep learning methods with high efficiency and lower data size requirements, which has more practical significance.

1. Introduction

Rotating machinery, an essential part of modern industry, is increasingly compact and integrated for a higher power-to-weight ratio. Along with terrible working conditions, fault symptoms become hard to detect, and fault patterns are challenging to recognize [1]. Developments of sensors, information transmission, and data analysis technology promote the application of data-driven fault detection and control [2], in which the pattern recognition model plays a vital role [3]. Vibration signal is widely used in the fault detection of rotating machinery because of easy acquisition and rich information, whereas the fault features are usually covered deeply in the one-dimensional time-dependent signal, resulting in the model's struggling balance between accuracy and efficiency. Therefore, it is necessary to research a superior method for rotating machinery fault detection.

Traditional fault detection is a step-by-step data processing approach including signal decomposition, feature extraction, feature selection, and mode identification. Vibration signals are commonly decomposed into components firstly. Then, features can be extracted and identified [4,5]. Machine learning algorithms such as support vector machine (SVM) [6] and fuzzy C-means (FCM) clustering [5] are widely used as classifiers but struggle in heterogeneous or nonconvex datasets. Therefore, Farhat et al. [7] employed the multi-kernel SVM (MSVM) model to analyze relevant features for bearing defects detection, and Oluwasegun et al. [8] combined the SVM and the K-means to diagnose anomalies in nuclear power plant components. Nonetheless, all of the above classifiers require features that fully describe the measured data. The applicability is also a challenge for signal decomposition and feature extraction and selection.

End-to-end fault diagnosis is proposed to simplify data preprocessed to improve applicability and efficiency. The data decomposition, feature extraction, and feature selection are canceled out or integrated into the classifier in the end-to-end approach. The raw data will be inputted into the classifier for pattern recognition directly, and the whole fault diagnosis contains only one step. Although the complexity is much reduced, the requirement for the classifier is increased. The information hidden in the one-dimensional signal calls for deep learning methods [9]. To this end, deep belief networks (DBNs) [10], convolutional neural networks (CNNs) [11], and recurrent neural networks (RNNs) [12] are employed in different applications. Topology adjustment and pretreatment are frequently used application-oriented optimization methods for artificial neural network (ANNs). Yan et al. [13] built a multiscale cascading DBN (MCDBN) to learn high-level features from multiscale characteristics of vibration signals parallel for rotating machinery faults detection. Deng et al. [14] proposed an improved quantum-inspired

[*] Corresponding author.
E-mail address: linjiewei@tju.edu.cn (J.W. Lin).

https://doi.org/10.1016/j.asoc.2022.109335
1568-4946/© 2022 Elsevier B.V. All rights reserved.

differential evolution (MSIQDE) to optimize hyper-parameters of the DBN for diagnosing bearing faults. On the other hand, Zhong et al. introduced a diversified DBN model based on pre-training and fine-tuning [15]. Compared with the DBN, CNN is more suitable for high-dimensional data [16]. The structure of CNN is critical to the implementation. The first optimization direction for CNN is adjusting the depth and size of the network, such as the CNN model based on LeNet-5 [17]. The second direction is to construct a parallel structure. Jiang et al. [18] designed a multiscale CNN (MSCNN) based on the multiscale coarse-grained layer to extract multiscale features and recognize gearbox faults from the vibration signal directly. The third one is to design particular convolution kernels. For example, Zhang et al. [19] used a CNN wide first-layer kernel (WDCNN) for bearing faults detection. However, the CNN with a complex structure does not necessarily have ideal performance because the feature maps may contain lots of irrelevant information. The attention mechanism (AM) can selectively ignore partial information and strengthen the other information to weight the features further. The most frequently used AMs are the channel attention mechanism (CAM) [20], the spatial attention mechanism (SAM) [21], and the convolutional block attention module (CABM) [22]. The first one weights features in the channel dimension, the second one weights features in the spatial dimension, and the third one combines the CAM and the SAM.

The RNN can deal with the time sequence better, but the standard RNN is difficult to train because of the gradient vanishing and exploding. As a variant of RNN, the long short-term memory (LSTM) network overcomes the above problems using the forget gate, the input gate, and the output gate by filtering information [23]. Based on the same theory, the gated recurrent unit (GRU) neural network is built using the reset gate and the update gate [24]. The LSTM and the GRU are well-matched in many applications. Optimization algorithms, such as artificial fish swarm (AFS) [25], and multilayer GRU (MGRU) [26] apply to these RNNs. Besides, pre-processing of the vibration signals using wavelet packet transform (WPT) [27] or ensemble empirical mode decomposition (EEMD) [28] also contributes to fault detecting rate. The data pre-processing is also combined with the CNN to improve local feature extraction. Zhou et al. [29] used CNN to mine the local spatial information of the partial discharge spectrum and LSTM to mine its time-series feature information, then proposed a CNN-LSTM model for partial discharge pattern recognition. Wang et al. [30] combined the advantages of the one-dimensional CNN in local features and the GRU in global and dynamic information to propose a CNN-GRU model for hybrid faults diagnosis. It is difficult to build a very deep structure in the LSTM and the GRU. Alternatively, independently RNN (IndRNN) provides another solution with independent neurons in the same layer and shows satisfying robustness [31]. However, training of deep learning model is highly dependent on the GPU performance because of the slow convergence speed caused by the back-propagation (BP) method, especially for the back-propagation through time (BPTT) in the RNNs. Only partial information is available in large-scale ANNs, making it even more resource-consuming for BP [32,33]. Besides, the local optimum brought by gradient descent is an adverse effect on the recognition rate. The over-fitting for small training datasets is hard to solve thoroughly, even using the sparse model built by dropout [10]. In practice, the fault detection of machinery requires simple hardware with low cost. It is one of the main reasons that widely accepted models such as the VGGNet [34] and the ResNet [35] have hardly been applied in engineering. In addition, requirements of the DBN and the RNNs for one-dimensional signals may destroy the spatial information of high-dimensional data. The characteristics of these algorithms are summarized in Table 1.

An ideal end-to-end faults detection method shall be constructed with a simple structure and trained by a small dataset to work efficiently and practically. As a novel type of RNN, the echo state network (ESN) [36] is attracting attention because it takes a randomly generated reservoir as the basic processing unit instead of hidden layer neurons, and its training process is a linear regression. The ESN shows great potential in many fields thanks to its simple structure and low data requirement, such as emotion recognition [37], vehicle faults diagnosis [38], and so on. However, the ESN has defects in mining deep information and dealing with spatial information. Many efforts have been made: Long et al. [39] designed a deep model stacked by multiple ESNs, and Wang et al. [40] combined the ESN with the DBN to increase the information mining depth. However, these models still struggle with complex structure and spatial information destroying during unfolding high-dimensional data. Ma et al. [41] collected all past echo states as the multi-time scale echo state representations and extracted their multiscale temporal dependencies by a convolutional layer for the ESN to propose a convolutional multi-time scale ESN model, whereas large training data is needed for high precision. Therefore, the development of an ESN model with a simple structure and small data requirement is essential for end-to-end faults detection.

In this paper, a deep ESN is proposed based on fixed convolution kernels. The main contributions are as follows:

(1) Fixed convolution kernels are designed based on the Prewitt filter, the Sobel filter, and the Gaussian lowpass filter for the spatial feature extraction without training. A one hidden layer autoencoder (AE) is built to reduce the dimensionality and extract features further to release the burden of the ESN and improve the generalization.

(2) An accurate and efficient model named FCK-DESN is proposed for end-to-end pattern recognition with the help of the fixed convolution kernels and the AE. Analyses prove that the model is advantageous to small datasets and robust to hyper-parameters, initial weights, and topology.

(3) The smoothed pseudo-Wigner–Ville distribution (SPWVD) is employed to provide rich time–frequency information for the FCK-DESN when dealing with the one-dimensional time-dependent vibration signal.

The content of this paper is organized as follows: Section 1 introduces the research background and significance. Section 2 gives the fundamental theories of algorithms. The deep ESN is proposed in Section 3. In Section 4, the bearing and diesel engine faults are recognized by the proposed deep ESN. Section 5 is the analysis and discussion. Conclusion and outlook are given in Section 6.

2. Background theories

2.1. Echo state networks

The ESN, a type of RNN, is a reservoir computing model, as shown in Fig. 1. The reservoir is an internal network that input signals can activate diverse states. These states describe features of the input signals through linear combination so that the output could be obtained by linear regression [36]. During training, only the output weights will be adjusted. It also avoids the local optimum, the vanishing gradient, and the exploding gradient brought by the gradient descent algorithm [37,38].

Supposing $u = \{u_1, u_2, \ldots, u_n\}$ is the input signal vector and $y = \{y_1, y_2, \ldots, y_m\}$ the output, the reservoir state $x = \{x_1, x_2, \ldots, x_N\}$ in leaky-integrator ESN is updated as:

$$x(t+1) = (1-\alpha\gamma)\, x(t) + \gamma f(W_{\text{in}} u(t+1) + W x(t) + W_{\text{back}}\, y(t)) \quad (1)$$

where α is the leaking rate, γ the gain, t the time-step, $x(t)$ the reservoir state in the tth update cycle, $f(\bullet)$ the activation

Table 1
Characteristics of algorithms.

Strategy	Method	Advantage	Disadvantage
Step-by-step	FCM [5]; SVM [6]; MSVM [7]; SVM-K-means [8].	Low requirement for the classifier.	High dependence on features; Low generalization; Complex process.
End-to-end	MSDBN [13]; MSIQDE-DBN [14]; Diversified DBN [15]. LeNet-5 [17]; MSCNN [18]; WDCNN [19]; CNN-CAM [20]; CNN-SAM [21]; CNN-CBAM [22]; VGGNet [34]; ResNet [35]. AFS-GRU [25]; MGRU [26]; WPT-LSTM [27]; EEMD-LSTM [28]; CNN-LSTM [29]; CNN-GRU [30]; IndRNN [31].	High efficiency and generalization; Low dependence on expert knowledge.	High requirement for training data and hardware; Complex structure; The DBN, the LSTM, the SAE, and the MLP may destroy the spatial information.

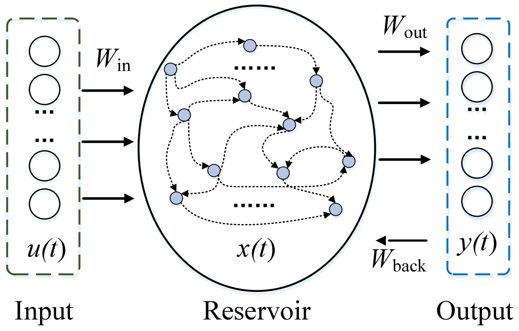


Fig. 1. Structure of the ESN.

function, W_{in} the randomly generated input weight matrix of $N \times n$, and W the randomly generated reservoir internal weight matrix of $N \times N$. N is the reservoir size, and n is the input size. W_{back} is the feedback weight matrix and usually neglected, and $\gamma = 1$ [37], so Eq. (1) can be transformed as:

$$x(t+1) = (1-\alpha)\,x(t) + f(W_{\text{in}}u(t+1) + Wx(t)) \tag{2}$$

The output can be computed as follows:

$$y(t) = g([x(t), u(t)]\,W_{\text{out}}) \tag{3}$$

where $g(\bullet)$ represents the activation function. W_{out} is the output weight matrix of $(N+n)\times m$, and the ESN is a multiple output classifier with m types of labels. The activation functions of the ESN are hyperbolic tangent (tanh) functions in this paper.

During the training process, only W_{out} is adjusted, whose objective function $L(\bullet)$ is:

$$L(\widehat{W}_{\text{out}}) = \left\| g^{-1}(y) - [x, u]\,W_{\text{out}} \right\|_2^2 \tag{4}$$

where $\|\cdot\|_2$ represents the L_2-norm. The estimate of the output weight matrix $\widehat{W}_{out}$ can be solved as:

$$\widehat{W}_{\text{out}} = [x, u]^{\dagger}\; g^{-1}(y) = ([x, u]^{\mathrm{T}}[x, u])^{-1}[x, u]^{\mathrm{T}}\; g^{-1}(y) \tag{5}$$

where the superscripts † and T represent the pseudo inverse and the transpose, respectively.

2.2. Convolution and pooling

This section describes the convolution and pooling operation processes and the feature map size to provide the theoretical basis for the following fixed convolution kernels design. Convolution can be regarded as feature extraction and dimensionality reduction, and pooling can be considered as down-sampling [16]. The convolution uses convolution kernels for traversing the input matrix to obtain the output. Supposing the input $D_{\text{in}} = \left[x_{ij}\right]_{h\times h}$ is a $h \times h$ matrix, the convolution kernel $K = \left[k_{ij}\right]_{d\times d}$ is a $d \times d$ matrix, and the output $C = \left[c_{ij}\right]_{m_c\times m_c}$ is a $m_c \times m_c$ matrix. Then,

$$m_c = (h + 2p_c - d)\,/s_c + 1 \tag{6}$$

where p_c is the size of padding and s_c the stride.

The element c_{ij} in the output matrix C is:

$$c_{ij} = f\left(\sum_{q=1}^{d}\sum_{l=1}^{d} x_{(s_c\times i+1)+q-1,(s_c\times j+1)+l-1} \bullet k_{ql} + b\right) \tag{7}$$

where b is the bias and $f(\bullet)$ the activation function.

The pooling layer is used to down-sample the convolutional data. The basic operation of pooling is characterizing a certain region by a specific value, normally the max-pooling or the mean-pooling. The pooling layer also ensures the rotation and translation invariances of the matrices to a certain extent.

3. Deep ESN based on fixed convolution kernels

To improve the capability of the ESN in mining deep information and dealing with spatial features, fixed convolution kernels are introduced to extract features, and the AE is used for dimensionality reduction. On this basis, a deep ESN based on fixed convolution kernels (FCK-DESN) is proposed.

3.1. Fixed convolution kernel

The ESN has high efficiency and requires small training data, whereas it is not good at dealing with spatial information. The CNN could extract local features gradually by convolutional layers and is suitable for high-dimensional data, but a large amount of data should train it. Especially for a deep network, the adjusting of weights by the BP algorithm is seriously slow. Thus, fixed convolution kernels are used for feature extraction. The procedure is replacing the trained convolution kernel K in Eq. (7) with the fixed convolution kernel.

Considering that the convolution kernel is essentially a filter, this paper designs fixed convolution kernels based on several classical filters. The priority of the fixed convolution kernel is to detect feature regions whose gradients are large in general. The edge detector is a common choice because it could eliminate irrelevant information and locate the energy concentration. The Prewitt filter and the Sobel filter are selected for simple structure and a certain image smoothing capability. Moreover, they are especially appropriate to be designed as large-size convolution kernels to extract features from single-channel images. The Prewitt filter can be regarded as an average filter and the Sobel filter as a weighted average filter. Their common forms are horizontal and vertical operators, as shown in Fig. 2(a)–(d). In particular, the Roberts operator, the Kirsch operator, and the Canny operator

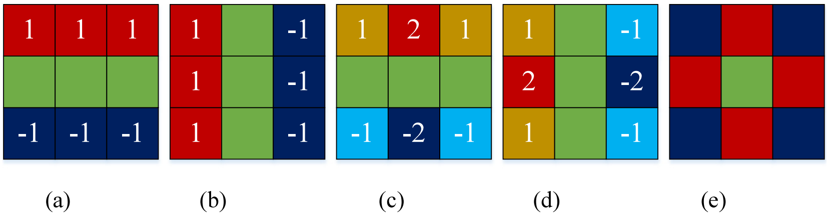


Fig. 2. Convolution kernels. (a) Horizontal Prewitt filter. (b) Vertical Prewitt filter. (c) Horizontal Sobel filter. (d) Vertical Sobel filter. (e) Gaussian lowpass filter.

are also frequently used edge detectors. However, the first two operators have formalized formworks. The Roberts operator is usually a matrix of 2 × 2, and the Kirsch operator is composed of eight matrices of 3 × 3. The Canny operator contains complex calculations and is not convenient to adjust based on the input signal. Compared with them, the Prewitt and the Sobel operators are universal and adaptable.

The Prewitt and the Sobel operators are simple filters and easily result in inaccurate edge location and rough edges, which require a filter to smooth and integrate the features. The Gaussian lowpass filter, see in Fig. 2(e), is used to reduce the Gaussian noise, smooth images, and fuse slight features. Supposing the size of the Gaussian lowpass filter is $d \times d$, then the element k_{ij} $(i, j \in [1, d])$ is:

$$k_{ij} = \exp\left\{-\left(D_i^2 + D_j^2\right)/2\sigma^2\right\} \tag{8}$$

where σ represents the standard deviation. D_i and D_j represent coordinate points of the filter in two directions, respectively. One can obtain $D_i = (2i - d - 1)/2$ and $D_j = (2j - d - 1)/2$, so:

$$k_{ij} = \exp\left\{-\left((2i - d - 1)^2 + (2j - d - 1)^2\right)/8\sigma^2\right\} \tag{9}$$

The σ is usually selected by the empirical formula $\sigma = d/3$. Considering that the big the σ, the smoother the result of the Gaussian lowpass filter, the σ is rounded up as $\lceil d/3 \rceil$ in this paper. When designing a convolutional layer, the size and number of kernels should be adjusted according to the actual condition. The Gaussian lowpass filter is set behind the Prewitt filter and the Sobel filter to avoid adverse effects on edge detection. The bias b in Eq. (7) is set linearly, such as $\{\cdots, -2, -1, 0, 1, 2, \ldots\}$.

3.2. Autoencoder

The disadvantage of the fixed convolution kernel is its low generalization capability. Besides, when extracting multi-scale features using several convolution kernels, the dimensionality of the processed data is hard to reduce. So, the AE is employed to compress the dimensionality and extract features further to release the burden of the ESN.

Since the deep AE with multilayer structure needs a large size of training data, a single hidden layer structure is chosen. Supposing the data processed by the convolutional layer is C,

$$\text{Encoder:} \quad u = f\left(CW_{\text{en}}^{\mathrm{T}}\right) \tag{10}$$

$$\text{Decoder:} \quad \widehat{C} = g\left(uW_{\text{de}}^{\mathrm{T}}\right) \tag{11}$$

where $\widehat{C}$ is the estimate of C, $f(\bullet)$ and $g(\bullet)$ are the activation functions.

The weight matrices, W_{en} and W_{de}, can be trained by the gradient descent algorithm. When the dimensionality of u is lower than that of C, an undercomplete AE can be obtained to improve the generalization capability and reduce the dimensionality.

3.3. FCK-DESN

Based on the fixed convolution kernels and the AE, features are extracted with a much smaller group of training data. Then the output weight matrix of the ESN, W_{out}, is trained. To activate the internal state of the reservoir, the processed signals are copied twice to obtain three of the same time-steps. The internal state computed by Eq. (2) could be unfolded as:

$$\begin{cases} x(1) = f(W_{\text{in}}u) \\ x(2) = (1 - \alpha)\,x(1) + f(W_{\text{in}}u + Wx(1)) \\ x(3) = (1 - \alpha)\,x(2) + f(W_{\text{in}}u + Wx(2)) \end{cases} \tag{12}$$

Special to note is that if there is only one time-step, Eq. (12) will be $x(1) = f(W_{\text{in}}u)$. The calculation ignores the reservoir, and the model is not an ESN. Furthermore, the reservoir size is usually small when the ESN is used for pattern recognition, leading to the information in $x(t)$ being compressed highly. Therefore, the input signal is also employed to ensure the information for the ESN. In this research, all the three internal states are collected by certain weights [36]:

$$\begin{cases} InSt\,(1) = 0.33\,[x(1), u] + 0.67\,[x(2), u] \\ InSt\,(2) = 0.67\,[x(2), u] + 0.33\,[x(3), u] \\ InSt\,(3) = [x(3), u] \end{cases} \tag{13}$$

The output in Eq. (3) could be computed as:

$$y = g\left([InSt\,(1), InSt\,(2), InSt\,(3)]\,W_{\text{out}}\right) \tag{14}$$

Supposing $InSt = [InSt\,(1), InSt\,(2), InSt\,(3)]$, the output weights could be computed as:

$$\widehat{W}_{\text{out}} = InSt^{\dagger}\,g^{-1}(y) = (InSt^{\mathrm{T}}InSt)^{-1}InSt^{\mathrm{T}}\,g^{-1}(y) \tag{15}$$

At this point, the FCK-DESN model can be constructed, as shown in Fig. 3. The model can be developed by three steps: (1) design the structure of the convolutional layer based on input data, (2) train the AE and obtain the low-dimensional data using the convolutional data, and (3) train the ESN. The fixed convolution layers and pooling are used to extract local features gradually without training. The encoder is mainly for generalization, and it can also reduce the dimensionality and extract features further. The ESN is finally employed to recognize the analyzed data for classification. The proposed FCK-DESN is given as **Model**.

Among the main hyper-parameters, the structure of the fixed convolutional layer containing the number of layers and channels, kernel size, and kernel type should be designed based on input data. Considering the deep layer with small kernel sizes is hard to control, the shallow layer with big kernel sizes is recommended in this paper. The n is also the dimensionality of encoder output in the AE. The leaking rate and spectral radius determine the characteristic of the ESN, and the reservoir size determines the model capacity. The research of Jaeger et al. [36] shows the α could be 0.2 in the classification task, and the other two, along with the n, will be analyzed in the following content.

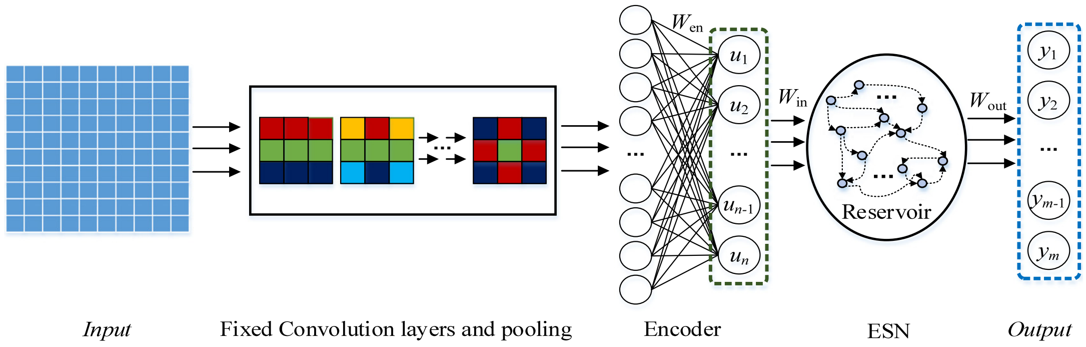


Fig. 3. Model of the FCK-DES.

Model: FCK-DESN

Input: Training data: $Data_{train}$ and $Label_{train}$, and testing data $Data_{test}$.
Output: Predicted labels of test data: $Label_{test}$
Hyper-parameters: Structure of fixed convolutional layer, learning rate and batch size of the AE, dimensionality of the ESN input n, leaking rate α, spectral radius ρ, and reservoir size N.
Training:
1. Based on Eq. (6), design the structure of fixed convolutional layer by the Prewitt, the Sobel, and the Gaussian lowpass filters.
2. Based on Eq. (7), perform convolution and pooling on $Data_{train}$ to obtain C_{train}.
3. Based on C_{train} and n, train the AE to obtain W_{en}. Compute u_{train} by Eq. (10).
4. Based on Eqs. (12)–(15), train W_{out} by u_{train} and $Label_{train}$.
Testing:
1. Use $Data_{test}$ to predict the labels of test data, $Label_{test}$, by Eqs. (7), (10), and (12)–(14).

The proposed model is trained block-by-block, in which the fixed convolution kernels do not need training, the AE has only one epoch training because it is just for dimensionality reduction and generalization capability, and the training of W_{out} is a linear regression. Supposing the dimensionality of input data for the AE is n_c and the number of neurons in the hidden layer is n, the multiply-accumulate operations (MACCs) of the AE could be computed as $2n_c \times n$ according to the encoder and decoder. Supposing the reservoir size of the ESN is N, the dimensionality of $InSt$ could be computed as $3(N+n)$ based on Eqs. (12)–(14). The MACCs of the training process in the ESN are $3(N+n)\times n_y$, where n_y is the dimensionality of output. Compared to traditional models, most parts of the FCK-DESN are fixed, and the output layer is trained by linear regression. Additionally, the model takes a spare reservoir as the basic processing unit. These are beneficial for obtaining an accurate and efficient model.

4. Experiments and results

Experiments of a rotor-bearing system and an engine test rig are employed to validate the capability of the proposed FCK-DESN in fault detection.

4.1. Data preparation

4.1.1. Bearing case

The bearing fault dataset from Case Western Reserve University[1] is used, including 7 conditions: the normal condition, the inner race defects of 0.007 and 0.014 in., the ball defects of 0.007 and 0.014 in., and the outer race defects of 0.007 and 0.014 in., respectively (hereinafter referred to as Conditions 0–6). The data is collected from the vibration acceleration sensor placed on the drive end under 1797 r/min with a sampling rate of 12 000 points per second. Taking the data of a single revolution as one sample, 300 samples per working condition and 2100 sample sets in total are obtained.

The frequency-domain information of the original vibration signal is hidden because it is a one-dimensional time series. The Wigner–Ville distribution (WVD) is a common time–frequency representation (TFR) algorithm, whereas it is bedeviled with quadratic cross-terms [42]. Several more bilinear time–frequency processing methods in the Cohen's class [43], including the Born–Jordan distribution (BJD), the Butterworth distribution (BD), the Choi–Williams distribution (CWD), the Margenau–Hill distribution (MHD), the Rihaczek distribution (RD), the pseudo-Wigner–Ville distribution (PWVD), and the smoothed pseudo-Wigner–Ville distribution (SPWVD) are used to analyze the vibration signal. These algorithms could be seen as the WVD with different kernel functions (details are shown in Appendix B) [44]. Meanwhile, smoothing windows in time and frequency domains have great influences on results. Hamming window can effectively reduce energy leaking because of the small side lobe, which is applicable for signals with complex spectrum [45]. Fig. 4 shows the time–frequency results of a signal in Condition 3 using the above algorithms with Hamming window. Besides, as representative TFR algorithms, short-time Fourier transform (STFT) and continuous wavelet transform (CWT) [44] are also studied (the wavelet coefficients are enlarged 20 times for unified scale).

The Renyi entropy is employed to compare these algorithms quantitatively (see the bottom right of Fig. 4). The smaller the Renyi entropy, the better the energy concentration degree [46]. The SPWVD is selected due to the lowest Renyi entropy of 5.89. Special to note is that the Renyi entropies of the BJD, the BD, and the CWD are similar to the SPWVD's, whereas this is not the research focus of this paper, so the SPWVD is selected just by a simple comparison.

The sample processed by the SPWVD is a 240 × 400 matrix, and 7 sets of signals in Conditions 0–6 (abbreviated as C0-6) are shown in Fig. 5, respectively. It contains rich information and a few cross-terms that will be taken as the input of the FCK-DESN.

4.1.2. Engine case

A bench test is performed on a turbocharged in-line diesel engine, as shown in Fig. 6(a). The engine is connected rigidly to a horizontal platform, which is supported by four air springs. The natural frequency of the air spring is below 2 Hz. The engine is driven by an electrical dynamometer. Acceleration sensors are placed on the Y-direction (the horizontal direction perpendicular to the crankshaft) of the engine block and the cylinder head cover.

[1] https://csegroups.case.edu/bearingdatacenter/home.

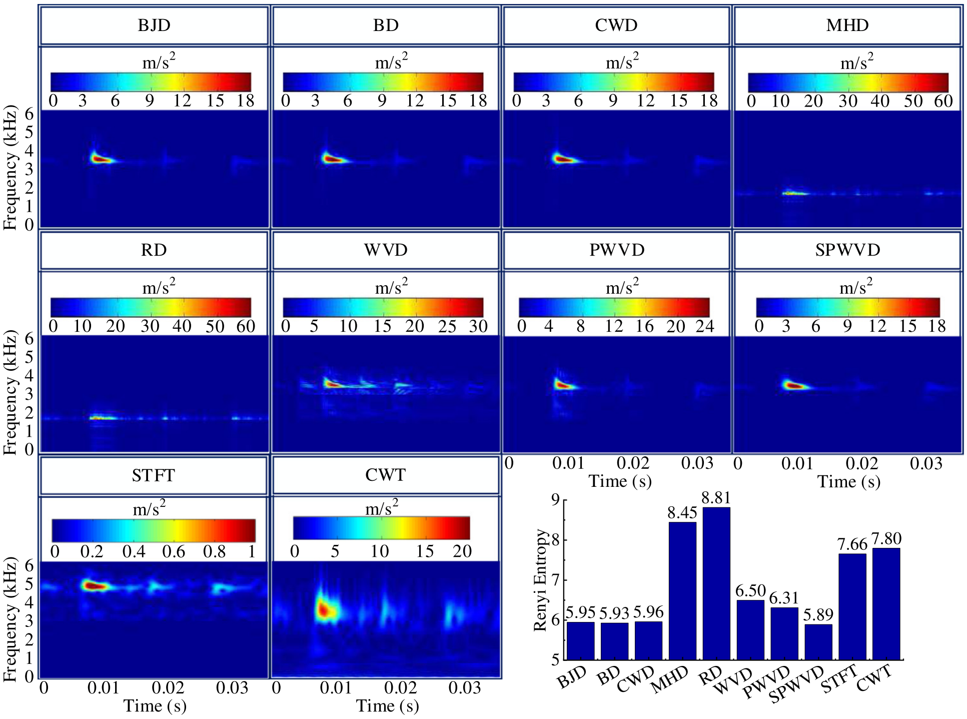


Fig. 4. Comparison of time–frequency analysis methods.

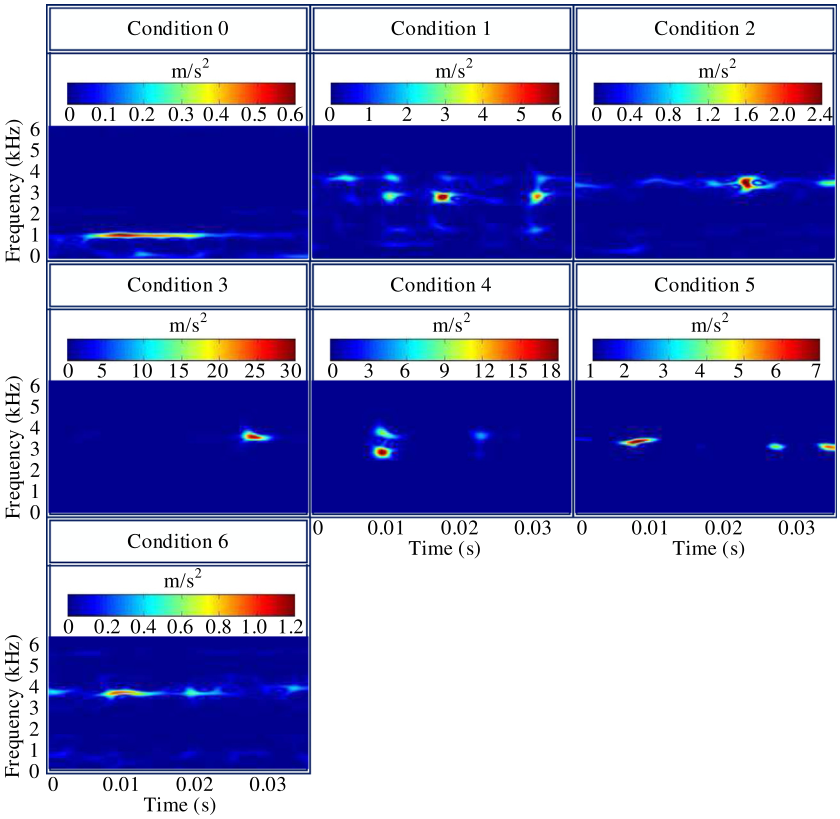
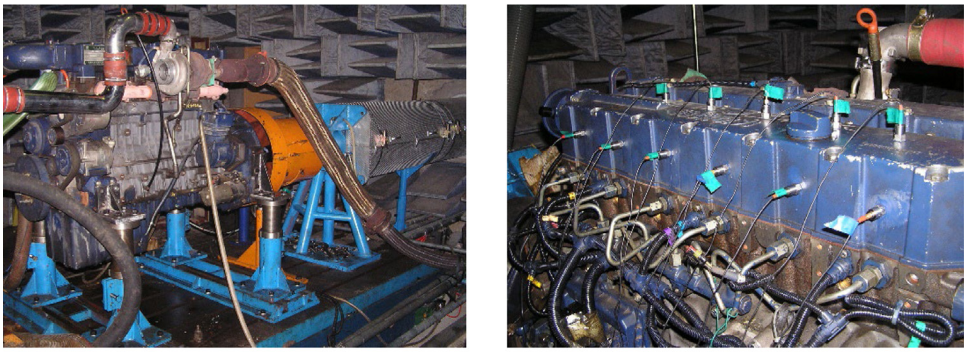


Fig. 5. Results of signal analyzed by the SPWVD.

As a reference, vibration in the Z-direction (the vertical direction) of the cylinder head cover is also measured, as shown in Fig. 6(b). Generally, the fault feature frequency of the tested engine is not higher than 10 kHz [47]. Based on the Nyquist sampling theory and the filtering characteristics of the testing equipment in cut-off frequency, the sampling rate is set as 25 600 points per second, i.e., the analysis frequency is 12.8 kHz. The sensor is the piezoelectric accelerometer ICP 621B40 (PCB, U.S.), whose frequency range is 1.6 Hz to 30 kHz. The signal is transmitted by coaxial cable PCB 002P30, whose shielding layer can avoid

Table 2
Dataset of engine faults.

Fault types	Description	Number of samples		
		1300 r/min	1600 r/min	Total
Normal working condition	Original condition	90	150	240
Abnormal fuel delivery	−25%	90	150	240
Delayed injection timing	+2 °CA	90	150	240
Advanced injection timing	−2 °CA	90	150	240
Big valve clearance	+0.05 mm	90	150	240
Small valve clearance	−0.05 mm	90	150	240
High rail pressure	+200 bar	90	150	240
Low rail pressure	−200 bar	90	150	240
Total	\	\	\	1920



(a) (b)

Fig. 6. Engine testing bench. (a) Testing engine. (b) Acceleration sensors.

electromagnetic interference. The data collector is the type SCM05 (Siemens, Germany), and its built-in filter can avoid aliasing interference. Thus, the equipment could ensure the reliability of testing.

According to [48], the four most frequent faults of diesel engines are reproduced in the experiments, including abnormal fuel delivery, abnormal injection timing, abnormal valve clearance, and abnormal rail pressure. The speeds during 1300 r/min-1600 r/min are economic operation range; thus, the maximum torque point at 1600 r/min and frequently used point at 1300 r/min are employed in this study. Certainly, comprehensive training data also could obtain the model for more working conditions. Besides, the physical performance of the working status in the experiment is the same as the practical application, which means the data is representative. The dataset is listed in Table 2, where "+" and "−" represent increasing and decreasing parameters from the normal condition, respectively, and CA the crankshaft angle.

The Y-direction data collected from the cylinder head cover near the third cylinder is analyzed. The vibration signal in one working cycle is taken as a sample, which means the crankshaft rotates two revolutions. After processed by the SPWVD, a 240 × 600 matrix is obtained (the spectra are not given because they are similar to Fig. 5). 240 sample sets in each working condition are selected, and 1920 sets are obtained in total.

4.2. Models and results

Based on the time–frequency domain data, two similar convolution structures containing three convolutional layers are designed for the bearing and engine faults detections, as shown in Fig. 7. For the bearing case, the first layer contains six 21 × 21 Prewitt filters composed of three horizontal and vertical operators. The second layer consists of six 11 × 11 Sobel filters, including three horizontal and vertical operators. The third layer contains three 11 × 11 Gaussian lowpass filters.

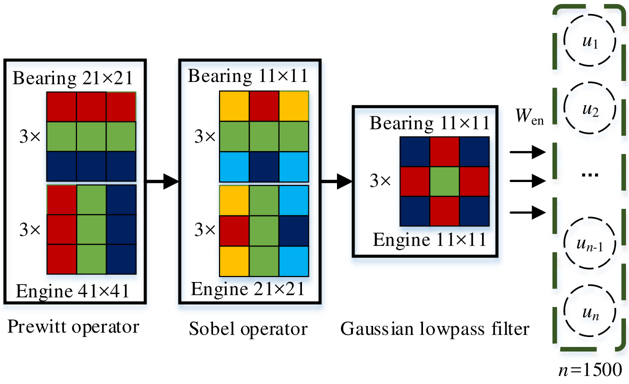


Fig. 7. Topologies of faults detection models.

Similarly, for the engine case, the first and second layers contain six 41 × 41 Prewitt filters and six 21 × 21 Sobel filters composed of three horizontal and vertical operators, respectively. The third layer contains three 11 × 11 Gaussian lowpass filters.

For each model, a 2 × 2 mean-pooling layer is added behind every convolutional layer. It is noted that the dimensionality of the ESN input (the encoder output of the AE) $n = 1500$, the reservoir size $N = 5$, and the spectral radius $\rho = 0.6$ are important hyper-parameters, which will be discussed later. An overall framework of the proposed method is shown in Fig. 8.

In the bearing case, 25 samples of every condition separately, 175 samples in total, are taken as the validation set, and 50 samples of every condition separately, 350 samples in total, are taken as the testing set. In the engine case, 20 samples of every condition separately, 160 samples in total, are taken as the validation set, and 40 samples of every condition separately, 320 samples in total, are taken as the testing set. Details are listed

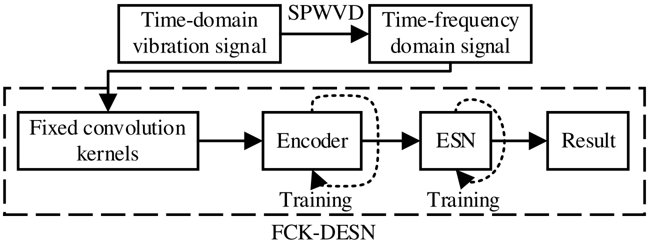

Time-domain vibration signal
SPWVD
Time-frequency domain signal
Fixed convolution kernels
Encoder
ESN
Result
Training
Training
FCK-DESN

Fig. 8. Overall framework of the proposed method.

Table 3
Divisions of the validation sets and testing sets.

Case	Bearing		Engine	
Dataset	Validation set	Testing set	Validation set	Testing set
Samples in every label	25	50	20	40
Number of labels	7		8	
Total	175	350	160	320

in Table 3. A parameter is defined as p = Training data size/All samples to analyze the influence of the training data size on the detection result under the same validation set and testing set. The dataset is divided in chronological order. The samples at the beginning are the training set, the samples in the middle are the validation set, and the samples at the end are the testing set. For comparison, the original ESN, the DBN, the CNN, the CNN-CAM, the CNN-SAM, the CNN-CBAM, the standard RNN, the LSTM, the GRU, and the IndRNN are also used. The best testing results out of ten runs of these algorithms are listed in Tables 4 and 5, considering that an ANN could give different results even with the same hyper-parameters due to its high sensitivity to initial weights. Besides, Recognition rate = $(Y_{\text{correct}}/Y_{\text{all}}) \times 100\%$, where Y_{correct} is the number of correctly detected samples, and Y_{all} is the number of all the samples.

Various original ESN and DBN models are tried: using them with different structures to test original time-domain signal and time–frequency domain data. Unfortunately, all of them, including sparse models, obtain unsatisfying results, and the best results recognized by time–frequency domain signal are shown. The recognition rates of the original ESN are all lower than 50%. The recognition rates of the DBN are 14.29% (1/7) for the bearing case and 12.50% (1/8) for the engine case, respectively. The reason is that these two models require one-dimensional input unfolded from the time–frequency domain signal, which destroys the spatial structure of the time–frequency matrix seriously. Besides, limited training data leads to over-fitting in the DBN. Several CNN models, including different convolutional layers and kernel sizes, are also tested, whereas they suffer the same problem and obtain the same recognition rates with the DBN (14.29% and 12.50%, respectively). Considering that the CNN could retain the spatial structure of the time–frequency matrix, several optimization methods such as dropout and batch normalization (BN) [49] are employed to alleviate the over-fitting. Based on the BN, two CNN models having the same structures with the convolutional layers of the FCK-DESN for the bearing and the engine cases separately (as shown in Fig. 7) show the best performances. Furthermore, three optimized models: the CNN-CAM, the CNN-SAM, and the CNN-CBAM, are also analyzed, whose results are listed in Tables 4 and 5. The CNNs could obtain recognition rates over 90% with large training data, whereas they decline sharply with decreasing p, which indicates the small training data is gradually unable to meet their requirements.

Considering the advantage in processing time series, the standard RNN, the LSTM, and the GRU models with three-layer structures are built to analyze the original vibration signal. The IndRNN with six layers is also tested. The RNN is unable to recognize the bearing and engine faults. The IndRNN has better recognition rates, whereas they are far from satisfactory. The recognition rates of the LSTM and the GRU in bearing case are about 90% but still lower than the FCK-DESN. The recognition rates of engine case are unsatisfying, especially for $p = 1/2$. The principal reason is that the engine vibration signal contains lots of impact components and noise, which would degrade the performances of the LSTM and the GRU seriously.

Besides, the training processes of the CNN, the standard RNN, the LSTM, the GRU, and the IndRNN heavily depend on the GPU, whereas they would take several days on the CPU. The FCK-DESN shows the best performance: it can maintain the accuracy over of 90% for all cases and holds steady in a certain with the p changing. Furthermore, the proposed method could identify different defects accurately with recognition rates over 80%. Based on the comprehensive analysis, the FCK-DESN has the highest classification precision and is advantageous in the requirement for training data at the same time.

5. Analyses and discussion

Some issues of the FCK-DESN are worthy of further discussion. The engine faults detection case ($p = 3/4$) is analyzed.

5.1. Hyper-parameters analyses

The dimensionality of the ESN input n, the reservoir size N, and the spectral radius ρ are important hyper-parameters in the FCK-DESN. The hyper-parameters are tested manually based on the validation set in this section. Firstly, the reservoir size and the spectral radius are set as $N = 4$ and $\rho = 0.4$ separately to analyze the dimensionality of the ESN input n, and recognition rates are listed in Table 6. The FCK-DESN can obtain the highest recognition rate of 98.75% in $n = 1500$ and $n = 1800$. The parameter is set as $n = 1500$ because a low dimensionality is conducive to efficiency.

Next, the reservoir size N and the spectral radius ρ are analyzed under $n = 1500$. The research of Jaeger et al. [36] shows that $\rho \in (0, 1)$ and $N < 10$ are appropriate for pattern recognition tasks in general. Recognition rates of different hyper-parameter combinations within these ranges are listed in Table 7. The results fluctuate around 98.00%, and several combinations can obtain the highest recognition rate of 99.38%. In this study, $N = 5$ and $\rho = 0.6$ are chosen because the small reservoir is more efficient. Results in Tables 6 and 7 indicate that the hyper-parameters have a certain influence on the accuracy, but not significantly. It means that the FCK-DESN is quite robust to these hyper-parameters when they are restrained in a reasonable range.

5.2. Significance test

As mentioned above, ANNs are sensitive to initial weights. Therefore, ten testing results of every model trained in random initial weights are analyzed. The p-values of t-tests between the FCK-DESN and the other ten models are listed in Table 8. In general, if the p-value is less than 0.05, the two results have significant differences. It shows the FCK-DESN is quite reliable, and its high accuracy does not benefit from random initial weights.

5.3. Cross-validation

As described in Section 4.2, the dataset is divided in chronological order. The first 180 samples in every condition separately

Table 4
Bearing faults recognition rates of different models.

Model	*p*	Normal working condition	Inner race defect		Ball defect		Outer race defect		Total
			0.007 in.	0.014 in.	0.007 in.	0.014 in.	0.007 in.	0.014 in.	
Original ESN	3/4	90.00%	30.00%	50.00%	26.00%	46.00%	22.00%	26.00%	41.43%
	2/3	90.00%	26.00%	48.00%	32.00%	32.00%	14.00%	26.00%	38.29%
	1/2	88.00%	10.00%	68.00%	20.00%	34.00%	12.00%	32.00%	37.71%
DBN	3/4	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	14.29%
	2/3	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	14.29%
	1/2	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	14.29%
CNN	3/4	100.00%	82.00%	90.00%	92.00%	94.00%	100.00%	92.00%	92.86%
	2/3	100.00%	80.00%	88.00%	98.00%	56.00%	100.00%	100.00%	88.86%
	1/2	100.00%	70.00%	78.00%	96.00%	58.00%	100.00%	96.00%	85.43%
CNN-CAM	3/4	100.00%	100.00%	100.00%	98.00%	60.00%	98.00%	100.00%	93.71%
	2/3	100.00%	100.00%	98.00%	92.00%	46.00%	96.00%	98.00%	90.00%
	1/2	96.00%	90.00%	90.00%	98.00%	40.00%	100.00%	94.00%	86.86%
CNN-SAM	3/4	100.00%	100.00%	98.00%	94.00%	56.00%	96.00%	98.00%	91.71%
	2/3	100.00%	100.00%	82.00%	96.00%	56.00%	94.00%	90.00%	88.29%
	1/2	100.00%	94.00%	96.00%	92.00%	30.00%	96.00%	92.00%	85.71%
CNN-CBAM	3/4	100.00%	100.00%	94.00%	92.00%	68.00%	100.00%	98.00%	93.14%
	2/3	100.00%	94.00%	92.00%	96.00%	58.00%	96.00%	98.00%	90.57%
	1/2	100.00%	94.00%	80.00%	92.00%	42.00%	98.00%	98.00%	86.29%
RNN	3/4	96.00%	38.00%	0.00%	2.00%	68.00%	94.00%	32.00%	47.14%
	2/3	96.00%	76.00%	0.00%	2.00%	46.00%	94.00%	10.00%	46.29%
	1/2	52.00%	28.00%	88.00%	16.00%	28.00%	0.00%	80.00%	41.71%
LSTM	3/4	100.00%	100.00%	84.00%	94.00%	70.00%	100.00%	98.00%	92.29%
	2/3	100.00%	76.00%	90.00%	96.00%	54.00%	100.00%	100.00%	88.00%
	1/2	98.00%	78.00%	86.00%	94.00%	44.00%	96.00%	98.00%	84.86%
GRU	3/4	100.00%	98.00%	100.00%	98.00%	54.00%	98.00%	98.00%	92.29%
	2/3	100.00%	100.00%	98.00%	80.00%	54.00%	100.00%	98.00%	90.00%
	1/2	100.00%	94.00%	78.00%	78.00%	50.00%	90.00%	96.00%	83.71%
IndRNN	3/4	90.00%	70.00%	66.00%	80.00%	70.00%	80.00%	68.00%	74.86%
	2/3	80.00%	68.00%	48.00%	60.00%	50.00%	64.00%	54.00%	60.57%
	1/2	64.00%	40.00%	54.00%	58.00%	40.00%	42.00%	44.00%	48.86%
FCK-DESN	3/4	100.00%	94.00%	96.00%	94.00%	84.00%	100.00%	100.00%	***95.43%***
	2/3	100.00%	88.00%	94.00%	94.00%	80.00%	94.00%	100.00%	***92.86%***
	1/2	100.00%	86.00%	88.00%	88.00%	80.00%	92.00%	100.00%	***90.57%***

are taken as the training set, the last 40 samples are the testing set, and the rest 20 samples are the validation set. The dataset is re-divided in this section for cross-validation. The first 40 samples are taken as the testing set in the first division, and so on. The testing results in the five new divisions are listed in Table 9. The recognition rates of the new divisions are all higher than 95%, which shows the proposed FCK-DESN does not benefit from the particular dataset.

5.4. Alternative models

As we know, the fixed convolution kernel does not require a large amount of training data, but its generalization capability is not satisfying. An alternative model can pre-train convolutional layers by the CNN and then implement the trained convolution kernels in the proposed algorithm. By this strategy, a testing result of 94.69% is obtained, which is lower than the FCK-DESN's. The reason is that the dataset size does not meet the training requirement of the convolutional layer. This approach may obtain a better result if the dataset is adequate.

Another alternative model is replacing the ESN with the LSTM or the GRU, i.e., FCK-LSTM or FCK-GRU. Unfortunately, they have the same testing result of 12.50% (1/8), which is the same as the DBN. Considering the BPTT algorithm used in the LSTM and the GRU, complex and limited training data is easy to result in serious over-fitting.

5.5. Optimization of ESN topology

Approaches that training W_{out} with the ridge regression [50] and optimizing the reservoir structure with the intrinsic plasticity (IP) [51] are usually used for large size reservoirs in time series predicting. They are used to optimize the pattern recognition model in this paper.

In the ridge regression, a regularization term is introduced to solve the ill-conditioned matrix. The objective function is changed as:

$$L(\widehat{W}_{\text{out}}, \lambda) = \left\| g^{-1}(y) - InStW_{\text{out}} \right\|_2^2 + \lambda \left\| W_{\text{out}} \right\| \tag{16}$$

It can be solved as:

$$\widehat{W}_{\text{out}} = InSt^{\dagger}\, g^{-1}(y) = \left(InSt^{\mathrm{T}} InSt + \lambda I\right)^{-1} InSt^{\mathrm{T}}\, g^{-1}(y) \tag{17}$$

where I represents the identity matrix and λ the regularization parameter.

The IP algorithm is used to improve the reservoir for information maximization [51]. The collected state $InSt$ is transformed into:

$$InSt_{\mathrm{IP}} = f\left(aInSt + b\right) \tag{18}$$

where $f(x) = 1/\left[1 + \exp(-x)\right]$. The a and b are parameters to be trained and they can be updated by:

$$\Delta b = \eta\left(1 - \left(2 + 1/\mu\right) InSt_{\mathrm{IP}} + InSt_{\mathrm{IP}}^2/\mu\right) \tag{19}$$

$$\Delta a = \eta/a + \Delta bInSt \tag{20}$$

where η is the learning rate and μ the average value of the desired output.

Unfortunately, the recognition rate of the FCK-DESN optimized by the ridge regression increases with decreasing regularization parameter λ. It exceeds 90.00% until $\lambda = 10^{-8}$ (90.31%). On the other hand, the testing result of the FCK-DESN optimized by the

Table 5
Engine faults recognition rates of different models.

Model	p	Normal working condition	Abnormal fuel delivery	Abnormal injection timing		Valve clearance failure		Abnormal rail pressure		Total
				+2 °CA	−2 °CA	+0.05 mm	−0.05 mm	+200 bar	−200 bar	
Original ESN	3/4	2.50%	20.00%	20.00%	17.50%	17.50%	12.50%	17.50%	7.50%	14.38%
	2/3	2.50%	10.00%	5.00%	20.00%	27.50%	12.50%	27.50%	15.00%	15.00%
	1/2	2.50%	12.50%	12.50%	12.50%	22.50%	12.50%	25.00%	12.50%	14.06%
DBN	3/4	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	12.50%
	2/3	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	12.50%
	1/2	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	12.50%
CNN	3/4	100.00%	100.00%	95.00%	97.50%	100.00%	97.50%	80.00%	75.00%	93.13%
	2/3	95.00%	100.00%	80.00%	100.00%	100.00%	100.00%	90.00%	47.50%	89.06%
	1/2	95.00%	100.00%	80.00%	95.00%	100.00%	85.00%	82.50%	50.00%	85.94%
CNN-CAM	3/4	100.00%	100.00%	80.00%	100.00%	100.00%	100.00%	90.00%	80.00%	93.75%
	2/3	95.00%	100.00%	75.00%	95.00%	100.00%	100.00%	85.00%	65.00%	89.38%
	1/2	100.00%	100.00%	75.00%	100.00%	100.00%	97.50%	82.50%	35.00%	86.25%
CNN-SAM	3/4	100.00%	100.00%	92.50%	95.00%	100.00%	95.00%	80.00%	75.00%	92.19%
	2/3	95.00%	100.00%	87.50%	95.00%	100.00%	97.50%	72.50%	70.00%	89.69%
	1/2	100.00%	100.00%	72.50%	92.50%	100.00%	95.00%	95.00%	37.50%	86.56%
CNN-CBAM	3/4	100.00%	100.00%	92.50%	97.50%	100.00%	95.00%	97.50%	70.00%	94.06%
	2/3	100.00%	100.00%	95.00%	80.00%	100.00%	95.00%	87.50%	70.00%	90.94%
	1/2	100.00%	100.00%	87.50%	70.00%	100.00%	92.50%	97.50%	47.50%	86.88%
RNN	3/4	32.50%	30.00%	0.00%	30.00%	10.00%	0.00%	10.00%	7.50%	15.00%
	2/3	12.50%	12.50%	10.00%	7.50%	10.00%	30.00%	12.50%	5.00%	12.50%
	1/2	30.00%	12.50%	5.00%	5.00%	5.00%	5.00%	15.00%	10.00%	10.94%
LSTM	3/4	100.00%	97.50%	87.50%	85.00%	95.00%	77.50%	82.50%	87.50%	89.06%
	2/3	100.00%	90.00%	82.50%	80.00%	67.50%	75.00%	72.50%	80.00%	80.94%
	1/2	85.00%	90.00%	60.00%	70.00%	85.00%	37.50%	57.50%	60.00%	68.13%
GRU	3/4	97.50%	95.00%	87.50%	82.50%	95.00%	82.50%	90.00%	87.50%	89.69%
	2/3	100.00%	95.00%	80.00%	70.00%	62.50%	65.00%	82.50%	87.50%	80.31%
	1/2	95.00%	97.50%	62.50%	42.50%	72.50%	47.50%	60.00%	52.50%	66.25%
IndRNN	3/4	80.00%	80.00%	77.50%	67.50%	62.50%	65.00%	80.00%	85.00%	74.69%
	2/3	82.50%	82.50%	75.00%	65.00%	42.50%	57.50%	60.00%	50.00%	64.38%
	1/2	75.00%	55.00%	40.00%	52.50%	57.50%	42.50%	40.00%	55.00%	52.19%
FCK-DESN	3/4	100.00%	100.00%	100.00%	90.00%	100.00%	97.50%	97.50%	100.00%	***98.13%***
	2/3	100.00%	100.00%	97.50%	97.50%	95.00%	97.50%	97.50%	87.50%	***96.56%***
	1/2	100.00%	100.00%	100.00%	92.50%	97.50%	95.00%	95.00%	90.00%	***96.25%***

Table 6
Recognition rates of the validation set in different ESN input dimensionalities.

n	1000	1100	1200	1300	1400	1500
Recognition rate	95.63%	96.88%	97.50%	96.25%	95.63%	98.75%
n	1600	1700	1800	1900	2000	
Recognition rate	95.63%	98.13%	98.75%	96.88%	97.50%	

Table 7
Recognition rates of the validation set in different reservoir sizes and spectral radii.

N / ρ	4	5	6	7	8
0.4	98.75%	97.50%	98.75%	99.38%	96.88%
0.5	98.13%	96.88%	98.13%	99.38%	98.75%
0.6	98.13%	99.38%	98.13%	97.50%	99.38%
0.7	98.75%	99.38%	96.88%	98.75%	98.75%
0.8	97.50%	98.75%	98.75%	96.25%	96.25%

IP algorithm is 98.13%, which is the same as the simple FCK-DESN. In other words, the optimization of the ESN topology is not necessary.

5.6. Comparison with classical method

A classical fault detection method of previous work is employed for comparison [47]. This approach uses the variational mode decomposition (VMD) to filter vibration signals from three channels firstly. Nine features are extracted from the filtered signal of each channel, respectively. Then, correlation-based feature selection (CFS) is taken as the fitness function of the genetic algorithm (GA) to search for the best feature subset from the redundant one. Finally, the expectation–maximization (EM) algorithm is used to cluster samples in an unsupervised way.

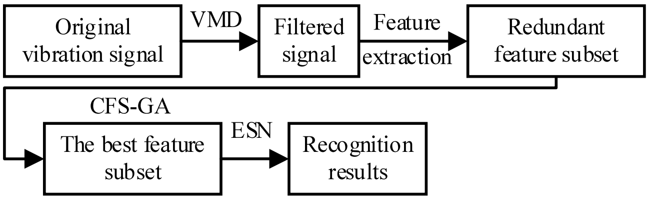


Fig. 9. Flow chart of the classical method.

For the current cases, 17 kinds of characteristic parameters are extracted from the signal filtered by the VMD: the minimum value, the maximum value, mean value, mean square value, root mean square, mean square error, standard deviation, kurtosis, margin, skewness, peak to peak, square root amplitude, average amplitude, fuzzy entropy, the Shannon entropy, the maximum singular value, and the fourth-order cumulant. The original ESN is taken as the classifier to ensure the credibility of the comparison. A flow chart of this classical method is shown in Fig. 9.

When detecting the bearing faults, features are searched using all 2100 sets of samples (300 sets in each condition). The best feature subset selected by the CFS-GA contains the mean value, the mean square value, the average amplitude, the Shannon entropy, and the maximum singular value. The same approach is

Table 8
P-values between the FCK-DESN and the other ten models.

Model	Original ESN	DBN	CNN	CNN-CAM	CNN-SAM
p-value	1.80×10^{-21}	3.49×10^{-24}	8.75×10^{-11}	1.52×10^{-10}	2.00×10^{-11}
Model	CNN-CBAM	RNN	LSTM	GRU	IndRNN
p-value	6.81×10^{-10}	5.29×10^{-19}	5.33×10^{-13}	3.42×10^{-13}	7.30×10^{-17}

Table 9
Recognition rates of five new divisions.

Case	1	2	3	4	5
Recognition rate	97.19%	96.25%	98.13%	98.44%	97.50%

Table 10
Bearing and engine faults recognition rates of the classical method.

Dataset	p	3/4	2/3	1/2
Recognition rate	Bearing	86.57%	85.71%	84.29%
	Engine	86.56%	85.63%	85.00%

employed to process the diesel engine vibration signals, and the best feature subset contains the margin, the skewness, the square root amplitude, and the Shannon entropy. The testing results are listed in Table 10, which are all lower than the results of the FCK-DESN listed in Tables 4 and 5.

The best feature subsets of the two cases are different, which means faults of various machines cannot be detected by the same method. Besides, the classical method consisted of signal decomposition, feature extraction, feature selection, and pattern recognition is a complex process with many influence factors. For example, the best subset of the bearing case selected from half of the samples (150 sets in each condition) contains the mean value and the square root amplitude. The best subset of the diesel engine case selected from half of the samples (120 sets in each condition) contains the mean value, the skewness, and the Shannon entropy. They are different from the best subsets selected from all the samples and prone to adverse effects on recognition rate. It is challenging to maintain each step ideal in the classical method, resulting in high dependence on the dataset. Moreover, it makes optimization and generalization difficult. Compared with it, the proposed FCK-DESN is advantageous because it could accomplish most of the work by deep learning capability to detect faults end-to-end.

6. Conclusions and outlook

A novel FCK-DESN is proposed to detect rotating machinery faults end-to-end with a small dataset. The fixed convolution kernels extract the features of data. The AE digs the deep information to improve generalization capability and reduce dimensionality. In the ESN, the processed data are copied twice to activate the internal states of the reservoir, and all three internal states are used to train the output weight matrix. Comparisons with the original ESN, the DBN, the CNN, the CNN-CAM, the CNN-SAM, the CNN-CBAM, the standard RNN, the LSTM, the GRU, and the IndRNN show that the FCK-DESN could always obtain the highest recognition rates in different datasets of bearing faults (95.43%, 92.86%, and 90.00%) and engine faults (98.13%, 96.56, and 96.25%).

In future work, more precise training methods with the small dataset for convolutional layers should be researched. The reservoir of the ESN is not fully understood. Developing the reservoir based on theoretical bases will also be the future direction.

Table A.1
List of acronyms.

Acronym	Description
DBN	Deep belief network
CNN	Convolutional neural network
CAM	Channel attention mechanism
SAN	Spatial attention mechanism
CABM	Convolutional block attention module
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
IndRNN	Independently recurrent neural network
ESN	Echo state network
AE	Autoencoder
FCK-DESN	Deep ESN based on fixed convolution kernels
SPWVD	Smoothed pseudo-Wigner–Ville distribution

Table A.2
List of notations.

Notation	Description
p	Training size/Testing size
α	Leaking rate
ρ	Spectral radius
N	Reservoir size of the ESN
n	Dimensionality of the ESN input
W	Reservoir internal weights of the ESN
$InSt$	Collection of internal states
C	Output of convolutional layer
W_{en}	Encoder weights

CRediT authorship contribution statement

Xin Li: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Visualization. **Fengrong Bi:** Resources, Writing – review & editing, Supervision. **Lipeng Zhang:** Resources. **Jiewei Lin:** Resources, Funding acquisition, Writing – review & editing, Supervision. **Xiaobo Bi:** Software. **Xiao Yang:** Investigation, 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.

Acknowledgments

This work was supported by the National Key R&D Program of China [Grant No. 2021YFD2000303].

Appendix A. Acronyms and notations

See Tables A.1 and A.2.

Appendix B. Several TFR algorithms in the Cohen's class

The Cohen's class time–frequency distribution of the continuous signal $z(t)$ is defined as [44]:

$$C_z^C(t,\omega) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A_z(\tau,\nu)\phi(\tau,\nu)\mathrm{e}^{-\mathrm{j}(\nu t+\omega\tau)}\mathrm{d}\tau\mathrm{d}\nu \tag{B.1}$$

Table B.1
Kernel functions of the algorithms.

The Cohen's class time frequency distribution	Kernel function
BJD	$[\sin(\tau\upsilon/2)]/(\tau\upsilon/2)$
BD	$1/\left[1+(\tau/\tau_0)^{2M}+(\nu+\nu_0)^{2N}\right]$
CWD	$\exp\left[-\alpha(\tau\nu)^2\right]$
MHD	$\cos(\tau\upsilon/2)$
RD	$\exp(\mathrm{j}\pi\tau\nu)$
WVD	1
PWVD	$h(\tau)$
SPWVD	$h(\tau)G(\nu)$

where t represents time, ω the frequency, τ the time delay, and ν the frequency offset.

The A_z is the ambiguity function of $z(t)$:

$$A_z(\tau,\nu)=\int_{-\infty}^{\infty} z(t+\tau/2)z^*(t-\tau/2)\mathrm{e}^{-\mathrm{j}\nu t}\mathrm{d}t \tag{B.2}$$

where the superscripts * represents conjugate matrix.

The $\phi(\tau,\nu)$ is the kernel function. The kernel functions of the algorithms mentioned in this paper are listed in Table B.1. where M represents the length of time, N the length of frequency, α the control parameter, and $h(\tau)$ and $G(\nu)$ the window functions. In this paper, $\alpha = 16$, $h(\tau)$ and $G(\nu)$ are both Hamming windows [46].

When the input is a discrete signal $z(n)$, its Cohen's class time–frequency distribution is:

$$C_z^D(n,\omega)=C_z'(n,\omega)*_n\otimes_\omega P(n,\omega) \tag{B.3}$$

where $*_n$ represents the convolution on n, $\otimes_\omega$ the cyclic convolution on ω, $C_z'(n,\omega)$ the time sampling function of the C_z^C.

$$P(n,\omega)=\sum_m \tilde{\phi}(n,m)e^{-\mathrm{j}m\omega} \tag{B.4}$$

where $\tilde{\phi}(n,m)$ represents the discrete kernel function with zero padding.

References

[1] R.N. Liu, B.Y. Yang, E. Zio, X.F. Chen, Artificial intelligence for fault diagnosis of rotating machinery: A review, Mech. Syst. Signal Process. 108 (2018) 33–47, http://dx.doi.org/10.1016/j.ymssp.2018.02.016.

[2] Y. Wang, Z.S. Wang, Data-driven model-free adaptive fault-tolerant control for a class of discrete-time systems, IEEE Trans. Circuits Syst.—II:Express Briefs 69 (1) (2022) 154–158, http://dx.doi.org/10.1109/TCSII.2021.3076890.

[3] J.D. Sun, C.H. Yan, J.T. Wen, Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning, IEEE Trans. Instrum. Meas. 67 (1) (2018) 185–195, http://dx.doi.org/10.1109/TIM.2017.2759418.

[4] X. Wen, G.L. Lu, J. Liu, P. Yan, Graph modeling of singular values for early fault detection and diagnosis of rolling element bearings, Mech. Syst. Signal Process. 145 (2020) 106956, http://dx.doi.org/10.1016/j.ymssp.2020.106956.

[5] X.Y. Bi, S.Q. Cao, D.M. Zhang, A variety of engine faults detection based on optimized variational mode decomposition-robust independent component analysis and fuzzy C-mean clustering, IEEE Access 7 (2019) 27756–27768, http://dx.doi.org/10.1109/ACCESS.2019.2901680.

[6] J.H. Shah, M. Sharif, M. Yasmin, S.L. Fernandes, Facial expressions classification and false label reduction using LDA and threefold SVM, Pattern Recognit. Lett. 139 (2020) 166–173, http://dx.doi.org/10.1016/j.patrec.2017.06.021.

[7] M.H. Farhat, X. Chiementin, F. Chaari, F. Bolaers, M. Haddar, Digital twin-driven machine learning: Ball bearings fault severity classification, Meas. Sci. Technol. 32 (4) (2021) 044006, http://dx.doi.org/10.1088/1361-6501/abd280.

[8] A. Oluwasegun, J.C. Jung, The application of machine learning for the prognostics and health management of control element drive system, Nucl. Eng. Technol. 52 (10) (2020) 2262–2273, http://dx.doi.org/10.1016/j.net.2020.03.028.

[9] X.X. Ding, Q.B. He, Energy-fluctuated multiscale feature learning with deep convnet for intelligent spindle bearing fault diagnosis, IEEE Trans. Instrum. Meas. 66 (8) (2017) 1926–1935, http://dx.doi.org/10.1109/TIM.2017.2674738.

[10] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507, http://dx.doi.org/10.1126/science.1127647.

[11] Y. LeCun, Y. Bengio, Convolutional Networks for Images, Speech, and Time Series, Holmdel, NJ, USA, 1998.

[12] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211, http://dx.doi.org/10.1207/s15516709cog1402_1.

[13] X.A. Yan, Y. Liu, M.P. Jia, Multiscale cascading deep belief network for fault identification of rotating machinery under various working conditions, Knowl.-Based Syst. 193 (2020) 105484, http://dx.doi.org/10.1016/j.knosys.2020.105484.

[14] W. Deng, H.L. Liu, J.J. Xu, H.M. Zhao, Y.J. Song, An improved quantum-inspired differential evolution algorithm for deep belief network, IEEE Trans. Instrum. Meas. 69 (10) (2020) 7319–7327, http://dx.doi.org/10.1093/ijlct/ctaa038.

[15] P. Zhong, Z.Q. Gong, S.T. Li, C.B. Schonlieb, Learning to diversify deep belief networks for hyperspectral image classification, IEEE Trans. Geosci. Remote 55 (6) (2017) 3516–3530, http://dx.doi.org/10.1109/TGRS.2017.2675902.

[16] B.L. Chen, H.D. Li, W.Q. Luo, Image processing operations identification via convolutional neural network, Sci. China-Inf. Sci. 63 (3) (2017) 139109, http://dx.doi.org/10.1007/s11432-018-9492-6.

[17] W. Long, X.Y. Li, L. Gao, Y.Y. Zhang, A new convolutional neural network-based data-driven fault diagnosis method, IEEE Trans. Ind. Electron. 65 (7) (2018) 5990–5998, http://dx.doi.org/10.1109/TIE.2017.2774777.

[18] G.Q. Jiang, H.B. He, J. Yun, P. Xie, Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox, IEEE Trans. Ind. Electron. 66 (4) (2019) 3196–3207, http://dx.doi.org/10.1109/TIE.2018.2844805.

[19] W. Zhang, G.L. Peng, C.H. Li, Y.H. Chen, Z.J. Zhang, A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals, Sensor 17 (2) (2017) 425, http://dx.doi.org/10.3390/s17020425.

[20] J. Hu, L. Shen, S. Albanie, G. Sun, E.H. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2023, http://dx.doi.org/10.1109/TPAMI.2019.2913372.

[21] X.Z. Zhu, D.Z. Cheng, Z. Zhang, S. Lin, J.F. Dai, An empirical study of spatial attention mechanisms in deep networks, in: 2019 International Conference on Computer Vision, Seoul, Korea, 2019, pp. 6687–6696, http://dx.doi.org/10.1109/ICCV.2019.00679.

[22] S. Woo, J. Park, J.Y. Lee, I.S. Kweon, CBAM: COnvolutional block attention module, in: 15th European Conference on Computer Vision, Munich, Germany, 2018, pp. 3–19, http://dx.doi.org/10.1007/978-3-030-01234-2_1.

[23] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[24] J. Chung, C. Gulcehre, K.H. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv:1412.3555.

[25] K. Zhao, H.D. Shao, Intelligent fault diagnosis of rolling bearing using adaptive deep gated recurrent unit, Neural Process. Lett. 51 (2020) 1164–1184, http://dx.doi.org/10.1007/s11063-019-10137-2.

[26] Y. Tao, X.D. Wang, R.V. Sanchez, S. Yang, Y. Bai, Spur gear fault diagnosis using a multilayer gated recurrent unit approach with vibration signal, IEEE Access 7 (2019) 56880–56889, http://dx.doi.org/10.1109/ACCESS.2019.2914181.

[27] J.F. Zhang, Y. Song, G. Li, C.Y. Wang, Y.F. Jiao, A method of fault diagnosis for rolling bearing of wind turbines based on long short-term memory neural network, Comput. Meas. Control 25 (1) (2017) 16–19, http://dx.doi.org/10.16526/j.cnki.11-4762/tp.2017.01.005.

[28] Y. Liu, H.P. Xu, N. Chu, F. Zheng, K.L. Wu, D.Z. Wu, Fan fault diagnosis based on long-short term memory network, J. Eng. Thermophys. 41 (10) (2020) 2437–2445.

[29] X. Zhou, X.T. Wu, P. Ding, X.G. Li, N.H. He, G.Z. Zhang, X.X. Zhang, Research on transformer partial discharge uhf pattern recognition based on cnn-lstm, Energies 13 (1) (2020) 61, http://dx.doi.org/10.3390/en13010061.

[30] Z.Z. Wang, Y.J. Dong, W. Liu, Z. Ma, A novel fault diagnosis approach for chillers based on 1-D convolutional neural network and gated recurrent unit, Sensors 20 (9) (2020) 2458, http://dx.doi.org/10.3390/s20092458.

[31] S. Li, W.Q. Li, C. Cook, C. Zhu, Y.B. Gao, Independently recurrent neural network (IndRNN): Building a longer and deeper RNN, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 5457–5466, http://dx.doi.org/10.1109/CVPR.2018.00572.

[32] G.Q. Tan, Z.S. Wang, Z. Shi, Proportional-integral state estimator for quaternion-valued neural networks with time-varying delays, IEEE Trans. Neural Netw. Learn. Syst. (2021) http://dx.doi.org/10.1109/TNNLS.2021.3103979.

[33] G.Q. Tan, Z.S. Wang, Reachable set estimation of delayed Markovian jump neural networks based on an improved reciprocally convex inequality, IEEE Trans. Neural Netw. Learn. Syst. (2021) http://dx.doi.org/10.1109/TNNLS.2020.3045599.

[34] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556v6.

[35] K.M. He, X.Y. Zhang, S.Q. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 771–778, http://dx.doi.org/10.1109/CVPR.2016.90.

[36] H. Jaeger, M. Lukosevivius, D. Popvici, U. Siewert, Optimization and applications of echo state networks with leaky- integrator neurons, Neural Netw. 20 (3) (2007) 335–352, http://dx.doi.org/10.1016/j.neunet.2007.04.016.

[37] F.J. Ren, Y.D. Dong, W. Wang, Emotion recognition based on physiological signals using brain asymmetry index and echo state network, Neural Comput. Appl. 31 (9) (2019) 4491–4501, http://dx.doi.org/10.1007/s00521-018-3664-1.

[38] O. Fink, E. Zio, U. Weidmann, Fuzzy classification with restricted boltzman machines and echo-state networks for predicting potential railway door system failures, IEEE Trans. Reliab. 64 (3) (2015) 861–868, http://dx.doi.org/10.1109/TR.2015.2424213.

[39] J.Y. Long, S.H. Zhang, C. Li, Evolving deep echo state networks for intelligent fault diagnosis, IEEE Trans. Ind. Inf. 16 (7) (2019) 4928–4937, http://dx.doi.org/10.1109/TII.2019.2938884.

[40] H.S. Wang, Q.M.J. Wu, J.B. Xin, J. Wang, H. Zhang, Optimizing deep belief echo state network with a sensitivity analysis input scaling auto-encoder algorithm, Knowl.-Based Syst. 191 (2019) 105257, http://dx.doi.org/10.1016/j.knosys.2019.105257.

[41] Q.L. Ma, E, H. Chen, Z.X. Lin, J.Y. Yan, Z.W. Yu, W.Y. Ng. Wing, Convolutional multitimescale echo state network, IEEE Trans. Cybern. 51 (3) (2021) 1613–1625, http://dx.doi.org/10.1109/TCYB.2019.2919648.

[42] Z.W. Liu, K. Wu, Z.S. Ma, Q. Ding, Vibration analysis of a rotating flywheel/flexible coupling system with angular misalignment and rubbing using smoothed pseudo Wigner–Ville distributions, J. Vib. Eng. Technol. 8 (2019) 761–772, http://dx.doi.org/10.1007/s42417-019-00189-y.

[43] L. Cohen, Generalized phase-space distribution functions, J. Math. Phys. 7 (5) (1996) 781–786, http://dx.doi.org/10.1063/1.1931206.

[44] X.D. Zhang, Modern Signal Processing, third ed., Tsinghua University Press, Beijing, China, 2015.

[45] M. Mottaghi-Kashtiban, M.G. Shayesteh, New efficient window function, replacement for the Hamming window, IET Signal Process. 5 (5) (2011) 499–505, http://dx.doi.org/10.1049/iet-spr.2010.0272.

[46] R.G. Baraniuk, P. Flandrin, A.J.E. Janssen, J.J. Michel, Measuring time-frequency information content using the Renyi entropies, IEEE Trans. Inform. Theory 47 (4) (2001) 1391–1409, http://dx.doi.org/10.1109/18.923723.

[47] X.B. Bi, J.S. Lin, F.R. Bi, X. Li, D.J. Tang, Y.X. Wu, X. Yang, P.F. Shen, Engine working state recognition based on optimized variational mode decomposition and expectation maximization algorithm, IEEE Access 8 (2020) 33545–33559, http://dx.doi.org/10.1109/ACCESS.2020.2975113.

[48] H.M. Nahim, R. Younes, H. Shraim, M. Ouladsine, Oriented review to potential simulator for faults modeling in diesel engine, J. Mar. Sci. Technol. 21 (3) (2016) 533–551, http://dx.doi.org/10.1007/s00773-015-0358-6.

[49] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv:1502.03167.

[50] J. Zhao, C. Zhao, F. Zhang, G. Wu, H.T. Wang, Water quality prediction in the waste water treatment process based on ridge regression echo state network, in: 2018 2nd International Conference on Artificial Intelligence Applications and Technologies, Shanghai, China, 2018, 012025, http://dx.doi.org/10.1088/1757-899X/435/1/012025.

[51] B. Schrauwen, M. Wardermann, D. Verstraeten, J.J. Steil, D. Stroobandt, Improving reservoirs using intrinsic plasticity, Neurocomputing 71 (7–9) (2008) 1159–1171, http://dx.doi.org/10.1016/j.neucom.2007.12.020.