Front. Mech. Eng. 2021, 16(4): 829–839
https://doi.org/10.1007/s11465-021-0652-4

RESEARCH ARTICLE

Jie LIU, Kaibo ZHOU, Chaoying YANG, Guoliang LU

Imbalanced fault diagnosis of rotating machinery using autoencoder-based SuperGraph feature learning

Abstract Existing fault diagnosis methods usually assume that there are balanced training data for every machine health state. However, the collection of fault signals is very difficult and expensive, resulting in the problem of imbalanced training dataset. It will degrade the performance of fault diagnosis methods significantly. To address this problem, an imbalanced fault diagnosis of rotating machinery using autoencoder-based SuperGraph feature learning is proposed in this paper. Unsupervised autoencoder is firstly used to compress every monitoring signal into a low-dimensional vector as the node attribute in the SuperGraph. And the edge connections in the graph depend on the relationship between signals. On the basis, graph convolution is performed on the constructed SuperGraph to achieve imbalanced training dataset fault diagnosis for rotating machinery. Comprehensive experiments are conducted on a benchmarking publicized dataset and a practical experimental platform, and the results show that the proposed method can effectively achieve rotating machinery fault diagnosis towards imbalanced training dataset through graph feature learning.

Keywords imbalanced fault diagnosis, graph feature learning, rotating machinery, autoencoder

1 Introduction

In industrial system, rotating machinery has been widely applied to many different kinds of machines [1,2]. The health state of rotating machinery affects the productivity of industrial production. However, the rotating machinery always works in extreme and complex environments. It would suffer from variable failures or damage, which may lead to machine breakdown and even cause casualties [3–5]. Therefore, it is necessary to monitor the health state of rotating machinery.

Recently, since the rapid development of data mining technique, the data-driven fault diagnosis methods have been widely applied. Data-driven fault diagnosis methods can effectively identify fault without too much prior knowledge using the deep learning or machine learning [6–8]. For example, Chen et al. [9] developed a diagnosis method using wavelet support vector machine. Chen et al. [10] proposed a convolutional neural network and cyclic spectral coherence-based diagnosis method. However, these data-driven methods rely on analysing monitoring signal to reveal internal failures, assuming that the sample size is approximately consistent for each health state [11]. Despite they can achieve good diagnostic performance on balanced dataset, the acquirement of balanced dataset in real industrial production is difficult [12]. In most cases, the imbalanced training datasets are always available. Performance of existing data-driven fault diagnosis methods would degrade when processing the imbalanced training datasets. Therefore, it is necessary to develop imbalanced fault diagnosis methods.

To overcome this limitation, various imbalanced fault diagnosis methods have been developed [13]. The strategy of making dataset balance is commonly used to process the imbalanced dataset. There are two main techniques to balance data, i.e., sampling method and data augmentation. The sampling methods basically can be divided into two categories, i.e., under-sampling and over-sampling. However, the under-sampling methods may lead to lose information of the majority classes, and the over-sampling methods tend to overlap samples and generate samples that do not provide useful information

Received March 26, 2021; accepted July 20, 2021

Jie LIU
School of Civil and Hydraulic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Kaibo ZHOU, Chaoying YANG (✉)
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
E-mail: yangcy@hust.edu.cn

Guoliang LU
School of Mechanical Engineering, Shandong University, Jinan 250061, China

[14]. In addition, the effect of the sampling methods depends on artificial settings. Differently, data augmentation-based methods can automatically select the optimal data enhancement scheme to generate data which can be judged as original data by means of deep learning. Deep learning not only can extract abstract feature from monitoring signals, bust also has shown promising effects in generating fake samples to balance dataset [15–17]. Among these deep-learning-based data augmentation methods, generative adversarial networks (GANs) can generate fake training samples which has the same data distribution as real training samples. Therefore, GAN-based data augmentation method has been widely applied to generate faker training samples. For example, Mariani et al. [18] proposed the balancing GAN, which can restore balance in imbalanced image datasets. Zhang et al. [19] adopted a deep GAN to generate fake training samples to expand dataset [19]. Gao et al. [20] proposed a Wasserstein GAN to generate training samples, and effectiveness of this method were verified in three industrial benchmark datasets. Actually, deep-learning-based data augmentation methods have showed the powerful ability of generating fake samples, but the generated fake training samples can never achieve the same training effect as the real training samples. In addition, the process of generating fake samples is computational costly. Therefore, it is of importance to develop a method that can overcome data imbalance without balancing data.

Construction of complex deep learning model is the commonly way to develop an imbalanced dataset fault diagnosis method without adopting data balance technique. For example, Zhao et al. [21] adopted Laplacian regularization technology to improve the deep autoencoder for imbalanced fault diagnosis of rotating machinery. Jia et al. [22] developed a deep convolutional neural network and improved weight normalization strategy for imbalanced fault classification. However, these methods ignore the relationship between samples. Relationship would vary with the change of machine health state, thus it would be an important indicator of fault diagnosis. In addition, constructing the relationship between samples could provide possibility to make full use of the unlabeled samples. Converting samples into graphs can help us construct the relationship between samples and make full use of the unlabeled samples.

Graph contains not only the extracted features from samples, but also the relationship between samples. There are many graph models in the fault diagnosis field, such as bond graph, periodic diagram and horizontal visibility graph [23–25]. These graph models are constructed by transforming raw data (RD) into graph data in a designed method. Compared to the RD, graph data can provide additional information, including not only the nodes value-based information, but also the relationships between nodes hidden in the geometry structure [26]. However, these graph model-based fault diagnosis methods cannot directly process the graph data to extract the graph feature, but convert the constructed graph into fault vectors for analysing. To some extent, it will lose partial information of the graph. As an alternative, the graph theory-based deep learning technique, graph convolutional network (GCN) can address this problem. The GCN can learn graph features from graph data to complete the classification problem in many fields, especially in fault field [27]. For example, Zhang et al. [28] applied a deep GCN model to bearing fault diagnosis firstly and demonstrated the performance of GCN. Wang et al. [29] proposed a graph that is constructed by the extracted vibration indicator, and the GCN was applied to learn the graph feature. Wang et al. [30] proposed a multiple micrograph-based GCN defect classification method, processing large-scale image datasets. Although these GCN-based graph feature learning methods have been successfully applied for machine fault diagnosis towards balanced dataset, there is a lack of research on the performance of GCN towards imbalanced training dataset.

Therefore, an imbalanced fault diagnosis of rotating machinery using autoencoder-based SuperGraph feature learning is proposed in this paper. The proposed method can make full use of limited training samples to improve diagnosis performance towards imbalanced training dataset, instead of generating fake training samples to keep dataset balance. The main contributions are highlighted as follows:

1) The extracted abstract features from unsupervised standard autoencoder can improve the quality of the constructed SuperGraph.

2) Graph theory is adopted to construct the SuperGraph using limited training samples, converting signal classification task into classical node classification problem.

3) GCN is utilized to achieve fault diagnosis for imbalanced dataset without generating fake training samples.

The rest of the paper is arranged as follows: Theoretical background about autoencoder, graph theory and GCN are briefly introduced in Section 2; the general procedure of proposed imbalanced fault diagnosis method will be described elaborately in Section 3; subsequently, the comprehensive experiments were conducted on two datasets, as shown in Section 4; finally, Section 5 concludes this paper.

2 Theoretical background

2.1 Autoencoder

As an unsupervised neural network, the autoencoder is composed of three layers, including encoder network,

hidden layer, and decoder network [31,32], as shown in Fig. 1. The encoder network is used to transform the high-dimensional input data into low-dimensional codes, which also can be regarded as features. The dimensionality reduction of the data is implemented in the hidden layer. The decoder network can reconstruct the input by minimizing the expected error between input and the output using the compressed codes.

The encoder network is explicitly defined as the encoding function denoted by e_θ [33]. For each given training input $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\}$, $\boldsymbol{x}^{(i)} \in R^m$, the encoder network is defined as follows:

$$\boldsymbol{h}^{(i)} = e_\theta\left(\boldsymbol{x}^{(i)}\right), \tag{1}$$

where $\boldsymbol{h}^{(i)} \in R^s$ is the encoded vector obtained from $\boldsymbol{x}^{(i)}$, and $s < m$.

The decoder network, reconstructing the input, is defined as a decoding function by $g_{\theta'}$. It maps $\boldsymbol{h}^{(i)}$ from low-dimensional space back into high-dimensional space, as shown in Eq. (2):

$$\boldsymbol{y}^{(i)} = g_{\theta'}\left(\boldsymbol{h}^{(i)}\right), \tag{2}$$

where $\boldsymbol{y}^{(i)} \in R^m$ is the reconstructed output.

The parameter sets of encoder and decoder are obtained while reconstructing the input by minimize the reconstruction error $L(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$, where $Loss(\cdot,\cdot)$ is a loss function that evaluates the discrepancy between input and reconstructed input [33].

In summary, the training of autoencoder aims at finding the parameter sets θ and θ' to minimize reconstruction error (ϕ_{AE}):

$$\phi_{AE}(\theta, \theta') = \frac{1}{m}\sum_{i=1}^{m} Loss\left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right). \tag{3}$$

The extensively used forms for the encoder and decoder are defined as follows [34]:

$$e_\theta(\boldsymbol{x}) = s_f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}),\ s_f(z) = \frac{1}{1+\mathrm{e}^{-z}}, \tag{4}$$

$$g_{\theta'}(\boldsymbol{x}) = s_g\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{d}\right),\ s_g(z) = \frac{1}{1+\mathrm{e}^{-z}}, \tag{5}$$

$$L\left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right) = \left\|\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right\|_2, \tag{6}$$

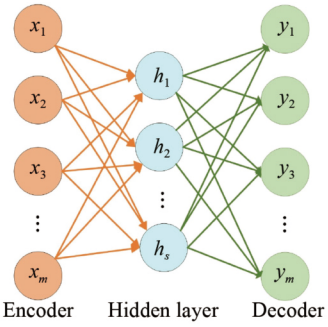

Fig. 1 Structure of autoencoder.

where s_f and s_g are the sigmoid function for encoder and decoder respectively. The parameter sets of the autoencoder are $\theta = \{\boldsymbol{W}, \boldsymbol{b}\}$ and $\theta' = \left\{\boldsymbol{W}^{\mathrm{T}}, \boldsymbol{d}\right\}$, where $\boldsymbol{b} \in R^s$ and $\boldsymbol{d} \in R^m$ are the bias vector, $\boldsymbol{W} \in R^{m\times s}$ and $\boldsymbol{W}^{\mathrm{T}} \in R^{s\times m}$ are the weight matrixes.

2.2 Graph representation

An undirected graph is denoted as $G = \{V, E, \boldsymbol{A}\}$. $V = \{V_i\}$ is the node set of graph, where $i = 1, 2, \ldots, N$, and N is the number of nodes; E represents the edge set; $\boldsymbol{A} \in R^{n\times n}$ is the adjacency matrix describing the edge connection, the element of $\boldsymbol{A}$ is 0 and 1. $A_{i,j} = 0$ represents there is no edge connection between nodes i and j, $A_{i,j} = 1$ represents there is edge between nodes i and j. An undirected graph containing four nodes and its corresponding adjacency matrix are shown in Fig. 2.

There are three definitions of the Laplacian matrix, i.e., the general, the symmetric normalized, and the random walk normalized. Among them, the symmetric normalized form is extensively used in spectral graph analysis, and the definition is shown in Eq. (7):

$$\boldsymbol{L} = \boldsymbol{I}_n - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}, \tag{7}$$

where $\boldsymbol{I}_n \in R^{n\times n}$ is the identity matrix, $\boldsymbol{D} \in R^{n\times n}$ is the diagonal degree matrix, and $\boldsymbol{L} \in R^{n\times n}$ is the Laplacian matrix. As a real symmetric matrix, orthogonal decomposition can be performed on Laplacian matrix $\boldsymbol{L}$, as shown in Eq. (9):

$$d_{i,j} = \sum_{j=1,\ j\neq i}^{N} a_{i,j}, \tag{8}$$

$$\boldsymbol{L} = \boldsymbol{U}\Lambda\boldsymbol{U}^{\mathrm{T}}, \tag{9}$$

where $\boldsymbol{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the eigenvalue matrix of Laplacian matrix $\boldsymbol{L}$, $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n)$ is the eigenvector matrix which is composed of eigenvectors $\boldsymbol{u}_i$.

2.3 Spectral convolution

Graph convolution includes spatial convolution and spectral convolution, in which spectral convolution has a clear mathematical explanation. Therefore, the spectral convolution based on graph Laplacian matrix is selected

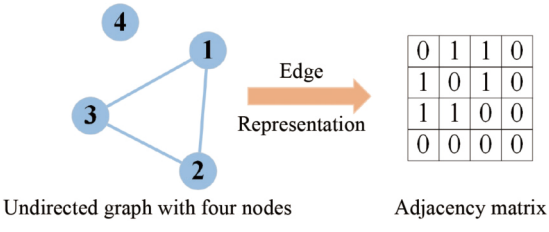

Fig. 2 An undirected graph containing four nodes and its corresponding adjacency matrix.

in this paper [35]. First, the filter operation of graph signal is shown in Eq. (10):

$$\boldsymbol{Y} = \boldsymbol{U}\varphi_\theta(\boldsymbol{\Lambda})\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}, \tag{10}$$

where $\boldsymbol{Y} \in R^{n\times s}$ represents the output of filter operation, $\boldsymbol{X} \in R^{n\times s}$ represents the graph signal, $\varphi_\theta(\cdot) = diag(\cdot)$ is a filter function, and $\varphi_\theta(\boldsymbol{\Lambda})$ can be understood as a function of eigenvalue of Laplacian matrix $\boldsymbol{L}$.

However, it is difficult and time-consuming to calculate $\varphi_\theta(\boldsymbol{\Lambda})$ [36]. The Chebyshev polynomial expansion is developed to overcome this limitation [37], as shown in Eq. (11):

$$\varphi_\theta(\boldsymbol{\Lambda}) \approx \sum_{k=0}^{K-1} \boldsymbol{\theta}_k T_k\left(\tilde{\boldsymbol{\Lambda}}\right), \tag{11}$$

where $\boldsymbol{\theta}_k$ is the Chebyshev coefficients, $T_k\left(\tilde{\boldsymbol{\Lambda}}\right)$ represents a function of the diagonal element of $\tilde{\boldsymbol{\Lambda}}$, and K is the Chebyshev polynomial coefficient. To be specific, $T_k(x)$ is a recursive calculation as demonstrated in Eq. (12):

$$T_0(x) = 1,\ T_1(x) = x,\ T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x),\ k \geqslant 2. \tag{12}$$

$\tilde{\boldsymbol{\Lambda}}$ is a normalized version of $\boldsymbol{\Lambda}$, as shown in Eq. (13):

$$\tilde{\boldsymbol{\Lambda}} = \frac{2\boldsymbol{\Lambda}}{\lambda_{\max}} - \boldsymbol{I}_n, \tag{13}$$

where $\lambda_{\max}$ is the largest element of $\boldsymbol{\Lambda}$, and the elements in $\tilde{\boldsymbol{\Lambda}}$ are in a range from –1 to 1.

Backing to the definition of a convolution of signal $\boldsymbol{X}$ with a filter φ_θ, the output $\boldsymbol{X}' \in R^{n\times m}$ of Chebyshev graph convolution layer finally can be calculated by

$$\boldsymbol{X}' = Cheb(\boldsymbol{X}, \boldsymbol{W}) = \sum_{k=0}^{K} \boldsymbol{\theta}_k \boldsymbol{U} T_k\left(\tilde{\boldsymbol{\Lambda}}\right)\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}', \tag{14}$$

where *Cheb* represents the function of Chebyshev graph convolution, and $\boldsymbol{W}' \in R^{s\times m}$ is a parameterized weight matrix.

3 Proposed imbalanced fault diagnosis method

In this paper, an imbalanced fault diagnosis of rotating machinery using autoencoder-based SuperGraph feature learning method is proposed. The general framework of the proposed imbalanced fault diagnosis method is introduced in Fig. 3. First, the traditional sensors are used to collect signals to monitor the rotating machinery operation state. Later, monitoring signals, acquired form the condition monitoring platform, are directly fed into the autoencoder without pre-processing. Meanwhile, a standard autoencoder is used to extract features from the input data. On the basis, the SuperGraph is constructed using the extracted features from autoencoder. Next, a GCN model is constructed to tackle fault classification

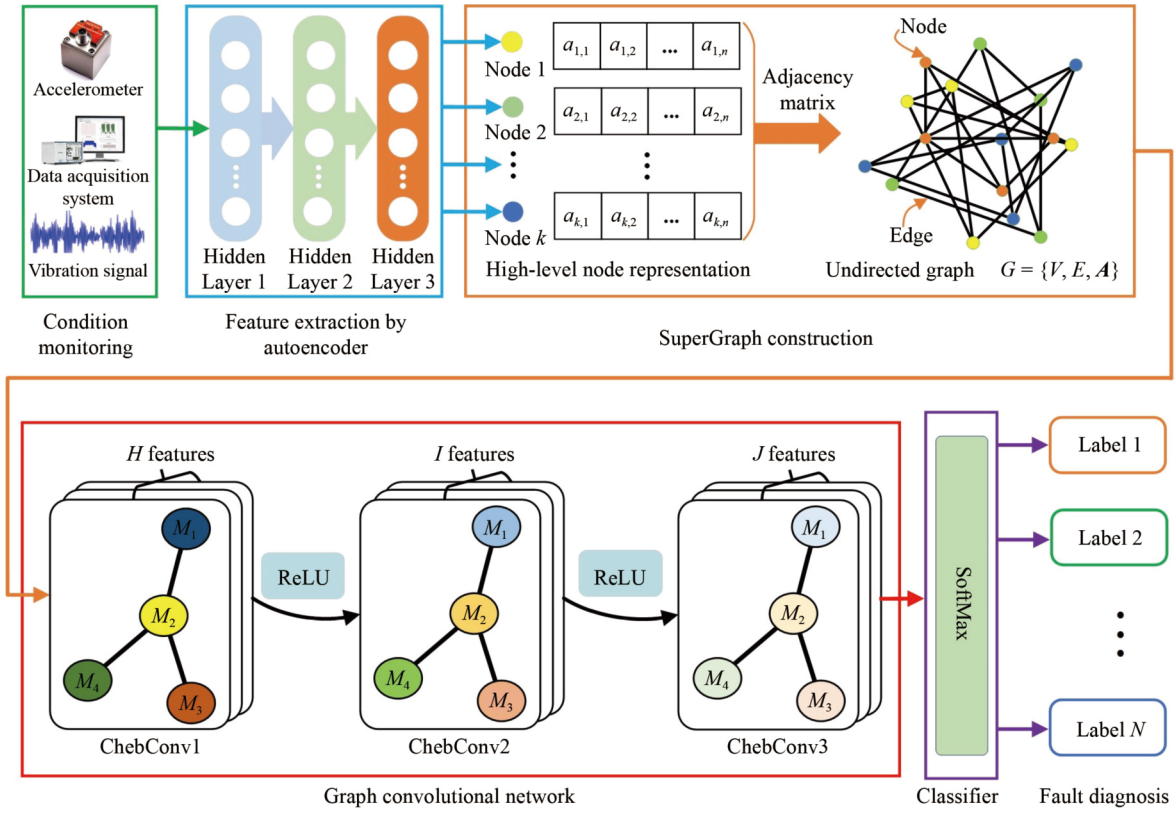

Fig. 3 General framework of the proposed imbalanced fault diagnosis method.

problem. Detailed description of the proposed imbalanced fault diagnosis method is illustrated below. In addition, the pseudo code of the algorithm is given as following:

Algorithm

Input: the monitoring signal $X=\{X_i\}$.
Output: the fault category Z.
1. Dividing $\boldsymbol{X}$ into $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{X}_{\text{other}}$;
2. Obtaining the high-level node representation: $\boldsymbol{h}^{(l)}=f_\theta\left(\boldsymbol{x}^{(l)}\right)$;
3. Obtaining the feature vector: $\boldsymbol{F}\leftarrow\boldsymbol{h}^{(l)}$;
4. Constructing the SuperGraph:
5. if $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in $\boldsymbol{X}_{\text{train}}$ and have the same fault;
6. $A_{ij}=1$
7. else if $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in $\boldsymbol{X}_{\text{other}}$ and both are collected at the same moment;
8. $A_{ij}=1$
9. else;
10. $A_{ij}=0$
11. Model training:
12. $Z\leftarrow GCN(X_{\text{train}})$;
13. loss;
14. updating with back propagation;
15. Output the fault category $\boldsymbol{Z}\leftarrow GCN(\boldsymbol{X})$;

3.1 Feature extraction

A given signal is denoted as $\boldsymbol{X}=\{x_1, x_2, \ldots, x_M\}$ with a number of M observation values. The constructed autoencoder which is trained in limited times, performs feature extraction on the input signal $\boldsymbol{X}$. The output of hidden layer $\boldsymbol{h}=\{h_1, h_2, \ldots, h_m\}$, where $m \leqslant M$, is the low-dimensional vector as the nodes attribute in the SuperGraph. After the training of the autoencoder, the reconstruction error became minimized. Finally, the optimal output of hidden layer was taken for the node attributes of monitoring signals.

3.2 Constructing the graph

As described in Section 2.2, the SuperGraph is denoted as $G=\{V, E, \boldsymbol{A}\}$. In the constructed SuperGraph, each node represents a monitoring signal, the attribute of nodes is their corresponding extracted feature, and the node connections rely on correlation between two nodes. An example of four types of signals and the corresponding SuperGraph are shown in Fig. 4. It can be seen that the SuperGraph is composed of four local graphs and all the nodes represent the monitoring signal. Each type of nodes represents each fault type of signals. For labeled nodes, the identical fault types of signals are interconnected. For unlabeled nodes, all samples obtained after sampling the data points collected at a certain time are regarded as having the same fault. On the basis, these obtained samples are interconnected. There, there are many local graphs in the SuperGraph, and all of the local graphs are independent in space without any edge connections. It is worth noting that the fault type of unlabeled nodes are unkown but identical.

After obtaining the SuperGraph, the input matrix of GCN model is composed of nodes attributes, and the process of input matrix transition is shown in Fig. 5. It can be seen that every row of the input matrix represents the feature vector of the corresponding node.

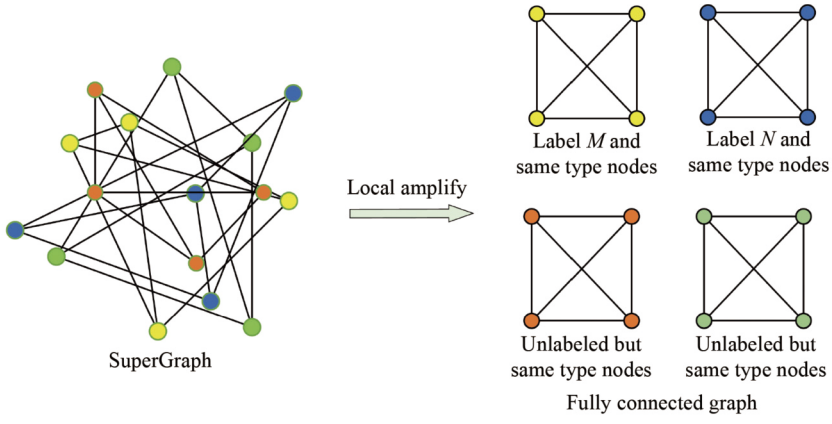

Fig. 4 SuperGraph with four types of nodes.

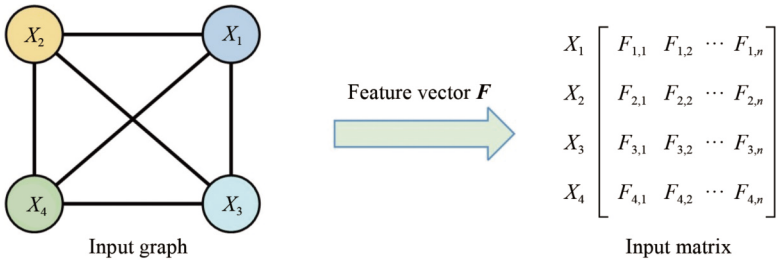

Fig. 5 Transition of input matrix.

3.3 A GCN model construction

According to the results that three-layer GCN model can effectively achieve diagnosis task [28], a GCN model with three layers of Chebyshev graph convolution, two layers of rectified linear unit (ReLU), and a softmax layer, was constructed in this paper. It can be simply described in Eq. (15):

$$\boldsymbol{Z} = softmax\left(Cheb\left(\sigma\left(Cheb\left(\sigma\left(Cheb\left(\boldsymbol{X},\boldsymbol{W}^{(1)}\right)\right),\boldsymbol{W}^{(2)}\right)\right),\boldsymbol{W}^{(3)}\right)\right), \tag{15}$$

where $\boldsymbol{Z} \in R^{N\times J}$ is the label of samples, $\boldsymbol{X} \in R^{N\times M}$ is the input matrix of graph signal, σ is the activation function of ReLU, and $\boldsymbol{W}^{(1)} \in R^{M\times H}$, $\boldsymbol{W}^{(2)} \in R^{H\times F}$, and $\boldsymbol{W}^{(3)} \in R^{F\times J}$ are the weight matrix of first, second, third Chebyshev convolution layer, respectively. H, F, and J represent the scales of input layer, first, second, third Chebyshev convolution layer, respectively, as shown in Fig. 6.

4 Experimental verification

To verify performance and effectiveness of the proposed imbalanced fault diagnosis method, comprehensive experiments were conducted on two datasets, including a bearing dataset from a Drivetrain Dynamic Simulator provided by Southeast University, China [38], and a practical experimental platform of gearbox [31]. All algorithms were performed in Python 3.8, Pytorch, and Pytorch Geometric which is a geometric deep learning extension library. They were processed by a server with an Intel Core i7-8700K CPU and a 32G RAM.

4.1 Experiments on a Drivetrain Dynamic Simulator

In this section, the experiments were conducted on a bearing dataset with five health states.

4.1.1 Experimental setup

The bearing dataset is acquired from the Drivetrain Dynamic Simulator, two working conditions that the load of rotating system is set to 20 Hz-0 V and 30 Hz-2 V are simulated [38]. There are five health states for the bearing data, including healthy, inner race fault, out race fault, ball fault, and combination fault on both inner race and outer race. There are 100 samples for each health state, and 50 samples for each working condition. The length of samples is 1024. The imbalanced Dataset A is originated from the constructed bearing dataset, the detailed setting of the bearing dataset and Dataset A is shown in Table 1.

4.1.2 Parameter setting

According to Ref. [39], the main parameter setting of the constructed autoencoder is shown in Table 2. In Ref. [28], three layers of GCN model achieves the best performance of fault diagnosis, and it also suggested that the structural parameter and kernel lengths have an impact on the performance of GCN model. Subsequently, identification of the proper parameters in GCN model is investigated in this paper. Four different experiments conducted in bearing dataset are applied in the parameter selection procedure to identify the structure of every convolution layer. In these experiments, 30% of samples in the bearing dataset are as training set, other 70% are as testing set. To avoid the contingency of the results, the average results of ten trials were used to analyse, and the results have been shown in Table 3. It can be seen that both the 25×16 and 30×25 can achieve the best classification results. However, training epoch of 30×25 is the less. Therefore, the hidden layer of GCN was set to 30×25. In summary, the scale of the input of GCN is equal to the size of feature vector, the scale of output is equal to the number of health states, and the scale parameter of hidden layer was set to 30×25.

As described in Section 2.3, the selection of will also affect the performance of GCN model. The kernel length K is the parameter in the Chebyshev graph convolution and its value directly affects the calculation result in feature learning phase. A comparison experiment was conducted on the bearing dataset by varying the kernel

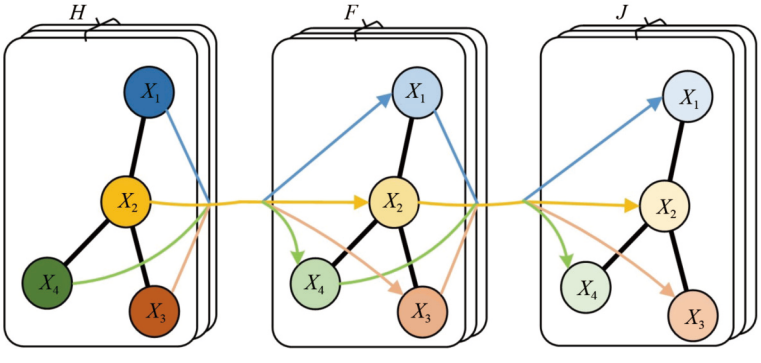

Fig. 6 Constructed GCN model.

length K to select the optimal value. The structure of GCN was set to 30×25. Results in Fig. 7 show that the classification accuracy is the highest when the kernel length K is from 4 to 5, and decreases as the kernel length K further increases. Finally, in this paper, the scale parameter of hidden layer was set to 30×25 and K was set to 4.

Table 1 Setting of the bearing dataset and Dataset A

Health state	Number of samples		Dataset A	
	Bearing dataset	Dataset A	Proportion of training samples/%	Proportion of testing samples/%
Health	100	100	50	50
Ball fault	100	70	20	50
Inner fault	100	80	30	50
Outer fault	100	60	10	50
Combination	100	60	10	50

Table 2 The parameter setting of the autoencoder

Parameter	Value
Learning rate	0.1
Number of hidden layers	3
Number of units in the input layer	1024
Number of units in the first hidden layer	512
Number of units in the second hidden layer	256
Number of units in the third hidden layer	50

Table 3 Performance of GCN in different scales of hidden layer on the bearing dataset

Hidden layer depth	Training accuracy/%	Testing accuracy/%	Epoch
30×16	100	99.43	59
25×16	100	99.71	36
20×16	100	98.57	33
30×25	100	99.71	52

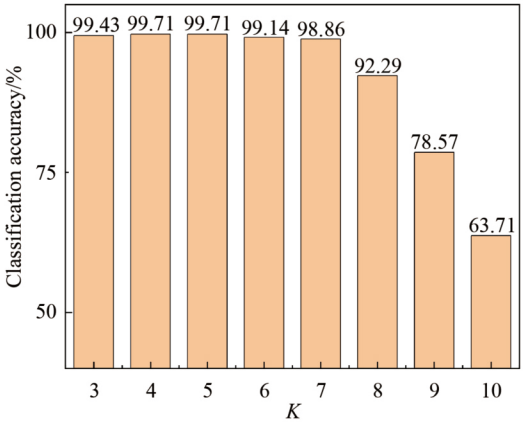

Fig. 7 Experimental results for different kernel lengths.

4.1.3 Diagnosis results analysis

In order to verify effectiveness of the proposed method, comparative experiments were conducted on Dataset A. Methods for comparison include convolutional neural network, vibration indicator-based GCN (VIG) [29], affinity graph-based GCN (AG) [40], and the detailed parameter setting of them is shown in Table 4. In the convolutional neural network (CNN), the RD was as the input. The statical features extracted from monitoring signals are used to construct graphs in the VIG. And the feature vector of AG is the half results of Fast Fourier transform (FFT). The experimental results of ten trials are shown in Table 5. It shows that the proposed fault diagnosis method can achieve the highest classification accuracy, reaching 98.27%. In addition, the cost of time for the proposed method is the least. Therefore, effectiveness of the proposed imbalanced fault diagnosis method is demonstrated.

4.1.4 Impact of selected features

In this paper, autoencoder-based SuperGraph feature learning technique is used for imbalanced data, partly improving the diagnosis performance. Despite without adopting a strategy to balance the data, the proposed fault diagnosis method can achieve good results towards imbalanced training dataset. Different from existed methods, the RD are converted into graph data so that more fault-related information can be obtained during features extraction.

Since the identical fault type of nodes has connections, the features of connected nodes are only composed of the corresponding fault information in the feature aggregation. Based on this node connection strategy, it greatly improves the ability to learn fault information while graph convolution is operated in the SuperGraph. Meanwhile, raw monitoring signals are compressed into the feature vector by the autoencoder, not only improving the recognition of features, but also reducing the volume of calculation for GCN feature aggregation. Therefore, the proposed method in this paper can identify faults well even towards imbalanced training dataset.

Table 4 Parameters setting of the comparative methods

Method	Setting
CNN	Input size: 370×1024, four layer of convolution, output size: (64, 1, 128), fully connected layer: (8192, 1)-(256, 1)-(5, 1)
VIG	Input size: 370×14, structure of GCN: 14-25-16-4, $K = 4$
AG	$2\beta^2 = 512$, $\varepsilon = 0.8$, input size: 370×512, structure of GCN: 512-200-100, fully connected layer: 300-4, $K = 1, 2, 3$
Proposed	Autoencoder: Table 2, structure of GCN: 50-30-25-11, $K = 4$

Further, to verify the impact of the selected feature to the GCN, comparison experiments were conducted on dataset A, where RD, half of the results of FFT, and traditional feature-based extraction (TF) are considered, including RD+GCN, FFT+GCN, TF+GCN, autoencoder-based extraction (AE)+GCN. Details of parameters setting are shown in Table 6. For all the comparison methods, the constructed graph is SuperGraph. In the TF+GCN method, a total of 29 time-domain and frequency-domain indicators as signal features, such as mean value, variance, mean frequency, variance frequency. $y(k)$ represents frequency spectrum of x, where $k = 1, 2, \ldots, N_K$. $x(n)$ represents data point of time-domain signal x with $n = 1, 2, \ldots, N$. f_k represents the frequency value of kth spectral line.

Ten trials were conducted to verify these methods, and the average results are listed in Fig. 8. It shows that RD+GCN achieve worst classification result, but using features extracted from signals as the feature vector could improve the diagnosis performance. Among these methods, the proposed method achieves the best classification result which verifies that the selected features will have an impact on diagnostic performance of GCN. Therefore, it is necessary to select the optimal features as the feature vector.

Table 5 Experimental results of the Dataset A

Model	Training accuracy/%	Testing accuracy/%	Time/s
CNN	100	88.65 ± 0.55	62
VIG	100	35.60 ± 0.64	45
AG	100	97.40 ± 0.32	53
Proposed	100	98.27 ± 0.32	42

Table 6 Parameters setting of comparison methods

Method	Setting
RD+GCN	Feature vector: unprocessed RD, hidden layer: 30×25, $K = 4$
FFT+GCN	Feature vector: half of results of FFT, hidden layer: 30 × 25, $K = 4$
TF+GCN	Feature vector: 29 time-domain and frequency-domain indicators, hidden layer: 30 × 25, $K = 4$

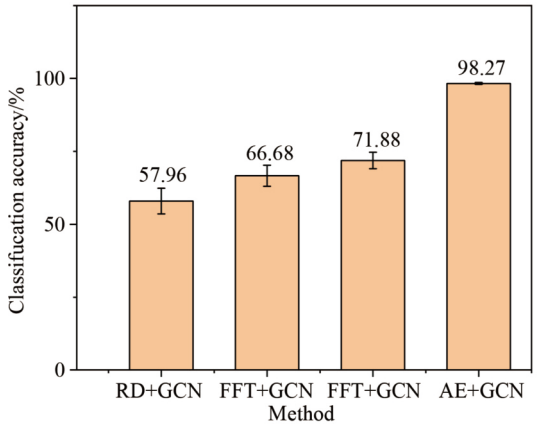

Fig. 8 Experimental results of different features extraction methods.

4.2 Experiments on a practical gearbox platform

In this section, the experiments were conducted on a practical gearbox experimental platform containing different crack severities.

4.2.1 Experimental setup

The practical experimental platform consists of a servo motor, a brake controller, a one-stage reduction gearbox, a torque sensor and a magnetic power brake [31], as shown in Fig. 9. Three triaxial accelerometers (PCB-356A16) were installed on different locations of the experimental platform. Driving gearbox contains four different radial crack lengths, i.e., 0, 5, 10, and 15 mm. The signal sampling frequency is 5 kHz, and the number of data points in each sample is 1024. The detailed information of practical experimental platform is listed as below:

Load: 0, 2, 4, 6, 8, and 10 N·m;
Crack length: 15, 10, 5, and 0 mm;
Input shaft speed: 1500 r/min;
Data points: 1024.

For the gearbox dataset, 20 monitoring samples under six different loads are randomly selected for each health state, thus a total of 480 samples are obtained for four health states. The imbalanced Dataset B is originated from the constructed gearbox dataset, the detailed setting of the gearbox dataset and Dataset B is shown in Table 7.

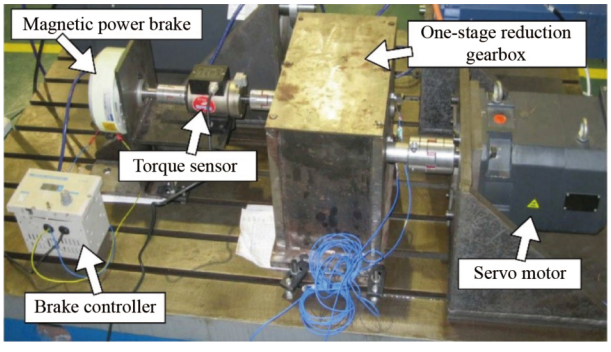

Fig. 9 Practical experimental platform for the one-stage gearbox.

Table 7 Information of the gearbox dataset and Dataset B

Crack length/mm	Number of samples		Dataset B	
	Gearbox dataset	Dataset B	Proportion of training samples/%	Proportion of testing samples/%
0	120	120	50	50
5	120	84	20	50
10	120	72	10	50
15	120	72	10	50

Table 8 Parameters setting of comparative methods

Method	Setting
CNN	Input size: 348×1024, four layer of convolution, output size: (64, 1, 128), fully connected layer: (8192, 1)-(256, 1)-(4, 1)
VIG	Input size: 1200×14, structure of GCN: 14-100-20, fully connected layer: 60-11, K = 1, 2, 3
AG	$2\beta^2 = 210$, $\varepsilon = 0.89$, input size: 1200×210, structure of GCN: 210-100-50, fully connected layer: 150-11, K = 1, 2, 3
Proposed	Autoencoder: Table 2, structure of GCN: 50-30-25-11, $K = 4$

Table 9 Experimental results of the Dataset B

Model	Training accuracy/%	Testing accuracy/%	Time/s
CNN	100	98.27 ± 0.27	66
VIG	100	88.63 ± 1.53	48
AG	100	93.38 ± 0.98	53
Proposed	100	99.08 ± 0.19	41

4.2.2 Diagnosis results analysis

The structure of the comparative method needs to be adjusted for Dataset B, the setting of comparison methods is shown in Table 8. Ten trails were conducted for each method, and the average results are used to analyse, as shown in Table 9. As shown in Table 9, the proposed fault diagnosis method can achieve 99.08% which is the highest among these methods. Besides, the standard deviation is 0.19% that shows the proposed method is more stable. Therefore, effectiveness of the proposed imbalanced fault diagnosis method is demonstrated.

5 Conclusions

In this paper, an imbalanced fault diagnosis of rotating machinery using autoencoder-based SuperGraph feature learning is proposed. The constructed SuperGraph reflecting the relationship between monitoring signals can help us make full use of a large number of unlabeled samples. By merits of powerful graph feature learning ability of GCN, the proposed method can effectively resist data imbalance. In addition, comparison experiments show that the proposed method can achieve the best diagnosis performance. Conclusions can be summarized: 1) Since the monitoring signals are converted to the SuperGraph, the relationship between signals can be extracted; 2) the proposed method can resist data imbalance without generating fake training samples; 3) the extracted abstract feature from autoencoder can improve the quality of the constructed SuperGraph.

Despite the proposed method can achieve excellent diagnosis performance, there are still problems. The constructed SuperGraph has redundant edges, such as all the labeled signals with identical fault type are interconnected, consuming excessive computation cost. The number of such edges can be reduced appropriately. In order to further improve the quality of constructed graph, many redundant edges should be cut. An improvement about few edges in SuperGraph would be considered in the future works.

Nomenclature

Abbreviations

AE	Autoencoder-based extraction
AG	Affinity graph-based GCN
CNN	Convolutional neural network
FFT	Fast Fourier transform
GAN	Generative adversarial network
GCN	Graph convolutional network
RD	Raw data
ReLU	Rectified linear unit
TF	Traditional feature-based extraction
VIG	Vibration indicator-based GCN

Variables

$\boldsymbol{A}$	Adjacency matrix
$\boldsymbol{b}$, $\boldsymbol{d}$	Bias vectors
$Cheb$	Function of Chebyshev graph convolution
$\boldsymbol{D}$	Degree matrix
E	Edge set
e_θ	Encoding function of autoencoder
f_k	Frequency value of kth spectral line
$\boldsymbol{F}$	Feature vector
$g_{\theta'}$	Decoding function of autoencoder
G	Undirected graph
H, F, J	Scales of graph convolution layer
$\boldsymbol{h}^{(i)}$	Output of hidden layer of autoencoder
$\boldsymbol{I}_n$	Identity matrix
K	Chebyshev polynomial coefficient
L	Reconstruction error
$\boldsymbol{L}$	Laplacian matrix
$Loss$	Loss function of autoencoder
m, M	Number of elements in vector
N	Number of nodes
s_f, s_g	Sigmoid function of encoder and decoder, respectively

T_k	Chebyshev polynomial
$\boldsymbol{u}_i$ (i = 1, 2, …, n)	Eigenvector
$\boldsymbol{U}$	Eigenvector matrix
V	Node set of graph
$\boldsymbol{W}$	Weight matrix
$\boldsymbol{W}'$	Parameterized weight matrix
$x(n)$	Data point of time-domain signal x
$\boldsymbol{X}$	Input signal
$\boldsymbol{X}'$	Output of Chebyshev graph convolution layer
$\boldsymbol{x}^{(i)}$	Input of encoder
$y(k)$	Frequency spectrum of x
$\boldsymbol{y}^{(i)}$	Output of decoder
$\boldsymbol{Y}$	Output of filter operation in GCN
$\boldsymbol{Z}$	Label of samples
σ	Activation function of ReLU
θ, θ'	Parameter sets of encoder
$\boldsymbol{\theta}_k$	Chebyshev coefficient
$\boldsymbol{\Lambda}$	Eigenvalue matrix of Laplacian matrix
$\tilde{\boldsymbol{\Lambda}}$	normalized version of $\boldsymbol{\Lambda}$
λ	Eigenvalue
$\lambda_{\max}$	Largest element of $\boldsymbol{\Lambda}$
φ_θ	Filter function of GCN
ϕ_{AE}	Reconstruction error of autoencoder

Acknowledgement This work was supported by the National Key R&D Program of China (Grant No. 2020YFB1711203).

References

1. Shao H, Jiang H, Wang F, et al. An enhancement deep feature fusion method for rotating machinery fault diagnosis. Knowledge-Based Systems, 2017, 119: 200–220
2. Han T, Liu C, Wu L, et al. An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems. Mechanical Systems and Signal Processing, 2019, 117: 170–187
3. Lei Y, Jia F, Lin J, et al. An intelligent fault diagnosis method using unsupervised feature learning towards mechanical Big Data. IEEE Transactions on Industrial Electronics, 2016, 63(5): 3137–3147
4. Chen Z, Mauricio A, Li W, et al. A deep learning method for bearing fault diagnosis based on cyclic spectral coherence and convolutional neural networks. Mechanical Systems and Signal Processing, 2020, 140: 106683
5. Han T, Li Y, Qian M. A hybrid generalization network for intelligent fault diagnosis of rotating machinery under unseen working conditions. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1–11
6. Guo X, Chen L, Shen C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. Measurement, 2016, 93: 490–502
7. Ren H, Liu W, Shan M, et al. A new wind turbine health condition monitoring method based on VMD-MPE and feature-based transfer learning. Measurement, 2019, 148: 106906
8. Razavi-Far R, Hallaji E, Farajzadeh-Zanjani M, et al. A semi-supervised diagnostic framework based on the surface estimation of faulty distributions. IEEE Transactions on Industrial Informatics, 2019, 15(3): 1277–1286
9. Chen F, Tang B, Chen R. A novel fault diagnosis model for gearbox based on wavelet support vector machine with immune genetic algorithm. Measurement, 2013, 46(1): 220–232
10. Chen Z, Mauricio A, Li W, et al. A deep learning method for bearing fault diagnosis based on Cyclic Spectral Coherence and Convolutional Neural Networks. Mechanical Systems and Signal Processing, 2020, 140: 106683
11. Mao W, He L, Yan Y, et al. Online sequential prediction of bearings imbalanced fault diagnosis by extreme learning machine. Mechanical Systems and Signal Processing, 2017, 83: 450–473
12. Dai X, Gao Z. From model, signal to knowledge: a data-driven perspective of fault detection and diagnosis. IEEE Transactions on Industrial Informatics, 2013, 9(4): 2226–2238
13. Zhang Y, Li X, Gao L, et al. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. Journal of Manufacturing Systems, 2018, 48: 34–50
14. Barandela R, Valdovinos R M, Sánchez J S, et al. The imbalanced training sample problem: Under or over sampling? In: Fred A, Caelli T M, Duin R P W, et al., eds. Structural, Syntactic, and Statistical Pattern Recognition. Berlin: Springer, 2004, 806–814
15. Guo L, Lei Y, Xing S, et al. Deep convolutional transfer learning network: a new method for intelligent fault diagnosis of machines with unlabeled data. IEEE Transactions on Industrial Electronics, 2019, 66(9): 7316–7325
16. Yang B, Lei Y, Jia F, et al. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. Mechanical Systems and Signal Processing, 2019, 122: 692–706
17. Li X, Zhang W, Ding Q. A robust intelligent fault diagnosis method for rolling element bearings based on deep distance metric learning. Neurocomputing, 2018, 310: 77–95
18. Mariani G, Scheidegger F, Istrate R, et al. BAGAN: Data Augmentation with Balancing GAN. 2018, arXiv:1803.09655
19. Zhang W, Li X, Jia X, et al. Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. Measurement, 2020, 152: 107377
20. Gao X, Deng F, Yue X. Data augmentation in fault diagnosis based on the Wasserstein generative adversarial network with gradient penalty. Neurocomputing, 2020, 396: 487–494
21. Zhao X, Jia M, Lin M. Deep Laplacian auto-encoder and its application into imbalanced fault diagnosis of rotating machinery. Measurement, 2020, 152: 107320
22. Jia F, Lei Y, Lu N, et al. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. Mechanical Systems and Signal Processing, 2018, 110: 349–367
23. Wang T, Liu Z, Lu G, et al. Temporal-spatio graph based spectrum analysis for bearing fault detection and diagnosis. IEEE

Transactions on Industrial Electronics, 2021, 68(3): 2598–2607
24. Lo C H, Wong Y K, Rad A B, et al. Fusion of qualitative bond graph and genetic algorithms: a fault diagnosis application. ISA Transactions, 2002, 41(4): 445–456
25. Wang T, Lu G, Yan P. A novel statistical time-frequency analysis for rotating machine condition monitoring. IEEE Transactions on Industrial Electronics, 2020, 67(1): 531–541
26. Gao Y, Yu D. Total variation on horizontal visibility graph and its application to rolling bearing fault diagnosis. Mechanism and Machine Theory, 2020, 147: 103768
27. Yang C, Zhou K, Liu J. SuperGraph: Spatial-temporal graph-based feature extraction for rotating machinery diagnosis. IEEE Transactions on Industrial Electronics, 2022, 69(4): 4167–4176
28. Zhang D, Stewart E, Entezami M, et al. Intelligent acoustic-based fault diagnosis of roller bearings using a deep graph convolutional network. Measurement, 2020, 156: 107585
29. Wang S, Xing S, Lei Y, et al. Vibration indicator-based graph convolutional network for semi-supervised bearing fault diagnosis. IOP Conference Series. Materials Science and Engineering, 2021, 1043(5): 052026
30. Wang Y, Gao L, Gao Y, et al. A new graph-based semi-supervised method for surface defect classification. Robotics and Computer-Integrated Manufacturing, 2021, 68: 102083
31. Liu J, Hu Y, Wang Y, et al. An integrated multi-sensor fusion-based deep feature learning approach for rotating machinery diagnosis. Measurement Science & Technology, 2018, 29(5): 055103
32. Tran V T, Althobiani F, Ball A. An approach to fault diagnosis of reciprocating compressor valves using Teager–Kaiser energy operator and deep belief networks. Expert Systems with Applications, 2014, 41(9): 4113–4122
33. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798–1828
34. Erhan D, Bengio Y, Courville A, et al. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 2010, 11: 625–660
35. Shuman D I, Narang S K, Frossard P, et al. The emerging field of signal processing on graphs: extending highdimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 2013, 30(3): 83–98
36. Song T, Zheng W, Song P, et al. Eeg emotion recognition using dynamical graph convolutional neural networks. IEEE Transactions on Affective Computing, 2020, 11(3): 532–541
37. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in Neural Information Processing Systems, 2016, 29: 3844–3852
38. Shao S, McAleer S, Yan R, et al. Highly accurate machine fault diagnosis using deep transfer learning. IEEE Transactions on Industrial Informatics, 2019, 15(4): 2446–2455
39. Shao H, Jiang H, Zhao H, et al. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. Mechanical Systems and Signal Processing, 2017, 95: 187–204
40. Li T, Zhao Z, Sun C, et al. Multi-receptive field graph convolutional networks for machine fault diagnosis. IEEE Transactions on Industrial Electronics, 2021, 68(12): 12739–12749