# Dynamic Graph-Based Feature Learning With Few Edges Considering Noisy Samples for Rotating Machinery Fault Diagnosis

Kaibo Zhou, Chaoying Yang, Jie Liu, *Member, IEEE*, and Qi Xu

***Abstract*—Due to its ability to learn the relationship among nodes from graph data, the graph convolution network (GCN) has received extensive attention. In the machine fault diagnosis field, it needs to construct input graphs reflecting features and relationships of the monitoring signals. Thus, the quality of the input graph affects the diagnostic performance. But it still has two limitations: 1) the constructed input graph usually has redundant edges, consuming excessive computational costs; 2) the constructed input graph cannot reflect the relationship between the noisy signals well. In order to overcome them, a dynamic graph-based feature learning with few edges considering noisy samples is proposed for rotating machinery fault diagnosis in this article. Noisy vibration signals are converted into one spectrum feature-based static graph, where redundant edges are simplified by the distance metric function. Edge connections of the input static graph are updated according to the relationship among high-level features extracted by the GCN. Based on this, dynamic input graphs are reconstructed as new graph representations for noisy samples. To verify the effectiveness of the proposed method, validation experiments were conducted on practical platforms, and results show that the dynamic input graph with few edges can effectively improve the diagnostic performance under different SNRs.**

***Index Terms*—Deep learning, dynamic graph (DG), fault diagnosis, graph convolutional network (GCN), rotating machinery.**

## NOMENCLATURE

### A. Sets

| | |
|---|---|
| $V$ | Set of nodes |
| $V_{\text{train}}/V_{\text{test}}$ | Set of training nodes/testing nodes |
| $E$ | Set of edges |
| $\Psi(V_p)$ | $k$ closest node neighbors of node $V_p$ |
| $Z$ | Set of health label |

### B. Parameters

| | |
|---|---|
| $G^0/G^d$ | Static graph/dynamic graph |
| $F/F^{\text{temp}}$ | Feature matrix/learned high-level feature matrix |
| $f$ | Node feature |
| $f_c/f_{\text{m}}$ | Samling frequency/meshing frequency |
| $X/X^{\text{nor}}$ | Raw original signal/normalized signal |
| $n$ | Number of samples |
| $T$ | Epoch |
| $M$ | Training epoch for a dynamic graph |
| $A$ | Adjacency matrix |
| $L$ | Laplacian matrix |
| $I_n$ | Identity matrix |
| $D$ | Degree matrix |
| $U$ | Eigenvector matrix |
| $\Lambda$ | Diagonal matrix of eigenvalues |
| $H$ | Output of spectral graph convolution |
| $g_\theta$ | Eigenfunction of $\Lambda$ |
| $T_k(x)$ | $k$th-order Chebyshev polynomials |
| $W$ | Trainable weight matrix |
| $Y$ | Output of graph convolution layer |
| $P_{\text{noise}}/P_{\text{signal}}$ | Power of noisy signal/raw signal |

### C. Abbreviations

| | |
|---|---|
| GCN | Graph convolution network |
| DL | Deep learning |
| CNN | Convolutional neural network |
| LSTM | Long short-term memory |
| ReLu | Rectified linear unit |
| $k$-NNG | $k$-nearest neighbor graph |
| FFT | Fast Fourier transform |
| SNR | Signal-to-noise ratio |
| ChebyNet | Chebyshev GCN |
| MRF-GCN | Multireceptive field GCN |
| VIG | Vibration indicator-based graph |
| AG | Affinity graph |
| SG | Static graph |
| DG | Dynamic graph |

Manuscript received April 9, 2021; revised July 29, 2021 and September 26, 2021; accepted October 9, 2021. Date of publication October 27, 2021; date of current version May 2, 2022. This work was supported by the National Key Research and Development Program of China under Grant 2020YFB1711203. *(Corresponding author: Jie Liu.)*

Kaibo Zhou, Chaoying Yang, and Qi Xu are with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zhoukb@hust.edu.cn; yangcy@hust.edu.cn; xuqi@hust.edu.cn).

Jie Liu is with the School of Civil and Hydraulic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: jie_liu@hust.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2021.3121748.

Digital Object Identifier 10.1109/TIE.2021.3121748

## I. INTRODUCTION

THE NORMAL operation of the rotating machinery is a prerequisite for guaranteeing machine productivity. It is

important to monitor the health state of the rotating machinery, avoiding major failure events and guaranteeing the reliability of the production process [1]–[3]. A large number of sensors are installed on the machine to monitor its operation state, providing a data basis for fault diagnosis. These sensor signals are processed and evaluated to judge whether faults have existed and then classify fault types.

Generally, the performance of the traditional fault diagnosis method relies on manually selected features [4]–[6]. Recently, machine learning-based diagnosis methods have been developed to automatically extract fault features from the monitoring signal without prior knowledge [7]–[9]. They can effectively deal with a small dataset but perform poorly when handling the ever-increasing big data.

The rapid development in computing power has spawned deep learning (DL)-based diagnosis methods, which can effectively process big data. DL models, such as autoencoder [10]–[12], convolutional neural network (CNN) [13]–[15], and long short-term memory (LSTM) [16]–[18], have been demonstrated an excellent technique for rotating machinery fault diagnosis. For example, Liu *et al.* [19] adopted a multisensor fusion strategy and an autoencoder for gearbox fault diagnosis. Li *et al.* [20] developed an automatic learning rate-based CNN for bearing fault diagnosis. Ma *et al.* [21] proposed an LSTM-based neural network, predicting the remaining life of rotating machinery. However, there are still some problems to be solved. These DL models can only learn the correlation between input signals, ignoring the relationships between them [22]. As the change of health states, these relationships of the signals vary a lot too. Thus, it is necessary to mine them for machine fault diagnosis.

To deal with this, the raw signals can be converted into graph data for further analysis [23]–[25]. From the node level, the signal is transformed into a node representation, and its features become node attributes. From the edge level, the edge connection between two nodes reflects the relationship between the signals. However, the traditional DL-based method just can operate on Euclidean structure data, including the image represented by the matrix and the signal represented by the vector. It cannot perform on non-Euclidean structure graph data [22].

Recently, graph convolution network (GCN) demonstrates its ability in learning the graph feature and accomplishing classification tasks in various fields [26]–[28]. For example, Marino *et al.* [29] used the GCN to learn the prior knowledge in the knowledge graph and successfully accomplished the image classification task. Yao *et al.* [30] successfully applied GCN to text classification tasks by constructing document relation networks and co-word networks. Kipf *et al.* [31] proposed a Chebyshev-polynomial-based GCN, classifying the corresponding categories of documents by learning the structural information in the citation network. The above results show the unique advantages of GCN in processing graph structure data and solving classification tasks, providing a reference for the application in the field of machine fault diagnosis. In essence, fault classification is also a classification task; therefore, GCN could have great potential in a machine fault diagnosis.

Motivated by successful cases of GCN, researchers begin to develop GCN-based fault diagnosis methods. However, raw signals are the Euclidean structure data, which need to be converted into non-Euclidean structure graph data reflecting features and relationships of the signals [32]–[34]. Therefore, a variety of graph construction methods are proposed for applying GCN to fault diagnosis. For example, Li *et al.* [35] adopted CNN to extract features for graph construction and used GCN for life prediction of rotating machinery. Wang *et al.* [36] proposed a vibration-indicator-based GCN, and the edge connections of the graph are determined by the Euclidean distance of vibration indicators. Yan *et al.* [22] developed an affinity-graph-based multireceptive field GCN, using the cosine similarity of frequency features to determine the edge connections. These graph construction methods relying on data points and time–frequency domain features have achieved good results, but there are still two limitations: 1) there are many redundant edges in the constructed graph, consuming excessive computational costs; 2) noise will make the signal analysis method invalid so that the constructed graph, relying on the signal features, cannot reflect the relationships between signals accurately.

To address these problems, a dynamic graph (DG)-based feature learning with few edges considering noisy samples is proposed for rotating machinery fault diagnosis in this article. Raw signals are converted into a static graph (SG) first using spectrum features extracted from noisy vibration signals, where the redundant edges are simplified by the distance metric function. The edge connections in the SG are only established between nodes and their $k$ closest neighbors. In the training process, the edge connections of the input SG are updated according to the relationship among high-level features extracted by the GCN. Based on this, dynamic input graphs are reconstructed as new graph representations for noisy samples. The main contributions are summarized as follows.

1) The redundant edges in the SG, which is constructed using the spectrum feature of a noisy sample are simplified by the distance metric function.
2) In the model training, the edge connections of the input graph are updated according to the relationship among high-level features extracted by the GCN.
3) The performance of GCN dealing with signals under different SNRs is analyzed and discussed.

The rest of this article is organized as follows. Section II introduces the principle of GCN. In Section III, the construction of the SG and the DG is presented. The proposed diagnosis framework is illustrated and summarized as an algorithm in Section IV. The effectiveness of the proposed method was verified by two case studies in Section V, and Section VI concludes this article.

## II. Theoretical Background

The spectral graph convolution and a standard GCN model are briefly described.

### A. Problem Description

Noise is the useless information component in the signal. Noise will reduce the ability of DL methods to extract fault features. Therefore, directly using the noisy signal as the input of the DL method cannot obtain a satisfactory diagnosis result.

In the existing research based on GCN, the relationship between samples is established by mining the preliminary characteristics of the signal, so as to obtain the graph [22]. The noise signal will greatly affect the structure of the graph and reduce the quality of the graph. The purpose of this article is to weaken the influence of noise on graph construction, improve the quality of graphs based on noisy signals, and improve the diagnosis performance of DL methods.

### B. Graph Convolution Network

For an undirected graph with node features, it can be represented as $G = \{V, E, A, F\}$, where $V = \{V_i\}_{i=1}^{n}$ denotes the node set. $E$ is the edge set, and $F \in R^{n\times m}$ denotes the feature matrix of graph $G$. $A \in R^{n\times n}$ represents the adjacency matrix, where the value can be defined as follows:

$$a_{i,j} = \begin{cases} 1, & \text{if nodes } V_i \text{and} V_j \text{ are connected} \\ 0, & \text{if nodes } V_i \text{and} V_j \text{ are unconnected.} \end{cases} \tag{1}$$

The symmetric normalized Laplacian matrix $L \in R^{n\times n}$ is defined as follows:

$$L = I_n - D^{-0.5} A D^{-0.5} \tag{2}$$

where $D \in R^{n\times n}$ is the degree matrix with $d_{i,i} = \sum_{j=1}^{n} a_{i,j}$, $I_n \in R^{n\times n}$ is the identity matrix.

Orthogonal decomposition is conducted on graph Laplacian matrix to obtain the eigenvectors and eigenvalues:

$$L = U \Lambda U^T \tag{3}$$

where $\Lambda = diag(\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$ consists of the eigenvalues and $U = [u_0, u_1, \ldots, u_{n-1}]$ is the eigenvector matrix.

The spectral graph convolution on graph $G$ with the feature matrix $F$ can be defined as follows:

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K-1} \theta'_k T_k(\tilde{\Lambda}) \tag{4}$$

$$H = \sum_{k=0}^{K-1} \theta'_k U T_k(\tilde{\Lambda}) U^T X \tag{5}$$

where $H \in R^{n\times m}$ is the output of spectral graph convolution, $U^T F$ is the graph Fourier transform, $g_\theta(\Lambda)$ is the eigenfunction of $\Lambda$ with a learnable parameter $\theta$, "$*$" denotes the graph convolution. $\tilde{\Lambda} = 2\Lambda/\lambda_{\max} - I_n$ consists of the rescaled eigenvalues, $\lambda_{\max}$ is the largest eigenvalues of $L$. $\theta'_k \in R^K$ is a vector of Chebyshev coefficients, $T_k(\tilde{\Lambda})$ is the $K$th-order Chebyshev polynomials, and the recurrence relation can be defined as $T_k(x) = 2xT_{k\text{-}1}(x)\text{-}T_{k\text{-}2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$. $K$ is the Chebyshev polynomial coefficient, which determines the receptive field of graph convolution.

With feature transformation, the final output $Y \in R^{n\times s}$ of the graph convolution layer can be obtained using a trainable weight matrix $W \in R^{m\times s}$ as follows:

$$Y = \text{Cheb}(X, W) = HW. \tag{6}$$

The standard two-layer GCN model for node classification [31] includes graph convolution layer, activation function, and classifier Softmax, as shown in Fig. 1:

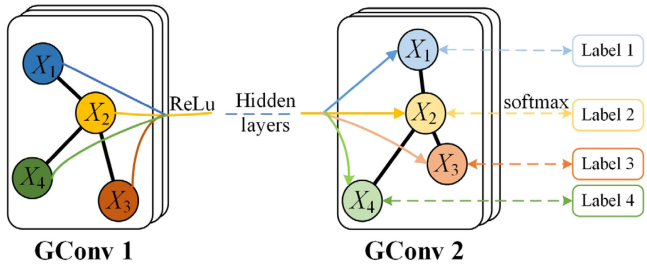


Fig. 1. Standard GCN model [34].

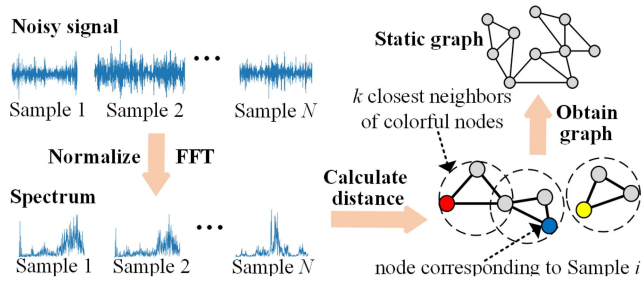


Fig. 2. Constructing the SG from samples.

$$Z = \text{softmax}\left(\text{Cheb}\left(\sigma\left(\text{Cheb}\left(X, W^{(1)}\right)\right)\right), W^{(2)}\right) \tag{7}$$

where $\sigma$ denotes the activation function, such as Rectified Linear Unit (ReLu), $W^{(1)} \in R^{m\times s}$, $W^{(2)} \in R^{s\times j}$ is the trainable weight matrix of first graph convolution layer and second graph convolution layer, $Z$ is the label of nodes.

## III. DG Model

### A. Constructing the SG From Spectrum

A $k$-nearest neighbor graph ($k$-NNG) with $k \geq 2$ has been proposed in the previous work [37]–[38] is a directed graph connecting each element to its $k$ closest neighbors. The $k$ closest neighbors of a node $p$ represent that the distance between nodes in the $k$ closest neighbors and node $p$ is the $k$th smallest distance from $p$ to other nodes. The samples of $k$ closest neighbors tend to share the same fault, and the edges between nodes with the same fault can enhance the fault feature learning ability of the GCN. However, since the Laplacian matrix of the directed graph is not symmetric, the spectral graph convolution cannot be conducted on a directed graph. Therefore, an SG is developed to turn the $k$-NNG into an undirected graph, as shown in Fig. 2. The following steps are performed to construct the SG.

***1) Data Preprocessing:*** For a signal $X = (x_1, x_2, \ldots, x_m)$, Min–Max normalization was adopted to obtain normalized $X^{\text{nor}}$ is shown in (8). In addition, $Z$-score normalization is also a commonly used data normalization method [35], and the effects of these two methods are verified by experiments to be similar

$$x_i{}^{\text{nor}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, i = 1, 2, \ldots, m \tag{8}$$

where the $X^{\text{nor}} = (x_1{}^{\text{nor}}, x_2{}^{\text{nor}}, \ldots, x_m{}^{\text{nor}})$ is the normalized signal.

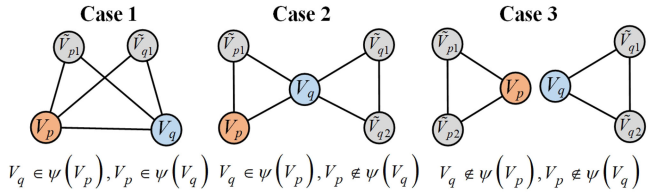


Fig. 3. Three cases of edge connection.

After data normalization, fast Fourier transform (FFT) is performed on each collected sample to obtain the spectrum as follows:

$$X' = \mathrm{FFT}(X) \tag{9}$$

where FFT(·) is used to obtain the frequency spectrum, $X' = (y_1, y_2, \ldots, y_m)$ is the results of FFT, and the half of results are taken as the sample node feature $f = (y_1, y_2, \ldots, y_{m/2})$.

***2) Finding the k Closest Neighbors:*** To obtain the SG, the $k$ closest neighbors of node $V_p$ need to be found using the Euclidean distance, which is a commonly used method to measure the absolute distance between samples [39]. This strategy of SG construction is similar to the operation in the $k$-nearest neighbor that calculates the similarity between samples to the cluster. The calculation is shown in follows:

$$\psi(V_p) = \left\{\tilde{V}_{pi}\right\}_{i=1}^{k}, \text{if } \mathrm{Dis}(fV_p, f\tilde{V}_{pi}) \text{is } k \text{ - th samllest} \tag{10}$$

where $fV_p$, $f\tilde{V}_{pi}$ are the node feature, Dis(·) is the Euclidean distance metric function, $\psi(V_p)$ is the $k$ closest neighbors set of node $V_p$, and $\tilde{V}_{pi}$ is the $k$ closest neighbor of node $V_p$.

***3) Constructing the SG:*** After finding the $k$ closest neighbors, the nodes in the $k$ closest neighbors of node $p$ are interconnected with node $p$ without direction. Repeating this operation for every node, the SG can be obtained. It is noted that node $q$ is a neighbor of node $p$ does not mean that node $p$ is a neighbor of node $q$. Therefore, the number of edges for every node is unsure, but at least ($k$-1). With the increase of $k$, the number of edges in the constructed graph increases, which leads to a longer calculation time of the algorithm. When the value of $k$ is too large, nodes with different labels from node $p$ are likely to appear in the $k$ closest neighbor set, which reduces the fault learning ability of the model and leads to a decline in the fault diagnosis performance. In this article, $k$ is artificially set to the minimum two according to verification experiments. Hence, by finding and connecting the $k$ closest neighbors of every node, the SG that consists of every sample can be obtained, and three cases for two nodes are shown in Fig. 3. The characteristics of SGs can be summarized as 1) each node has at least two edges while limiting the total number of edges to prevent edge redundancy; 2) the edge connections between nodes with the same label can be established without knowing the label of each node.

## B. Constructing the DG

Noise will change the $k$ closest neighbors of nodes in the SG so that two nodes with low similarity would be connected. Therefore, in order to improve the quality of noisy sample-based SGs, a DG is constructed as shown in Fig. 4. The steps are as follows.

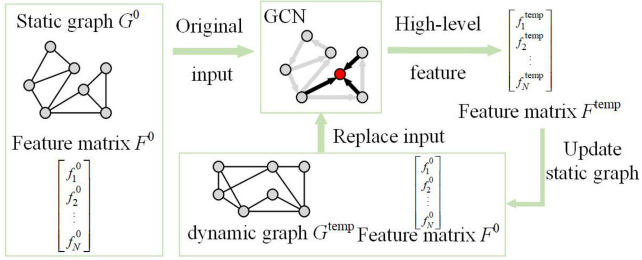


Fig. 4. Constructing the DG.

*Step* 1: The input SG is used to train the GCNs of cross-entropy loss for several epochs.

*Step* 2: The high-level features extracted by the trained GCN are taken as the reference features of the node, and the edge connections of the input SG are updated according to these features.

*Step* 3: The DG is constructed during the process of training, and it is the updated SG. The DG is regarded as a new input graph for the training of the GCN.

*Step* 4: After several times of updates, the final updated DG $G^n$ is fed into the trained GCN model to obtain labels of every node.

The number of nodes in the DG has not changed and is the number of samples. As for the edges, they have been updated by using the new feature matrix $F^{\text{temp}}$ to reconstruct the relationship between samples.

# IV. Proposed Diagnosis Framework

The overall flowchart of the proposed diagnosis framework, including data process, DG construction, and fault diagnosis using GCN, is shown in Fig. 5. The corresponding algorithm is given as follows.

# V. Experimental Verification

To verify the effectiveness of this proposed method, experiments were conducted on two datasets including a one-stage reduction gearbox and a planetary gearbox. All the algorithms were written in Python and Pytorch with Intel Core i7-8700K CPU.

## A. One-Stage Reduction Gearbox

***1) Data Description:*** As shown in Fig. 6, the experiment platform is composed of a brake controller, a magnetic power brake, a torque sensor, a one-stage reduction gearbox, and a servomotor. The accelerometers (PCB-356A16) were placed on the one-stage reduction gearbox to collect vibration signals. In addition, the shaft speed was set to 1500 and 900 r/min, so meshing frequency $f_{\mathrm{m}}$ can be calculated as 1250 and 750 Hz. The sampling frequency $f_{\mathrm{c}}$ was set to 5 kHz ($f_{\mathrm{c}} >= 2f_{\mathrm{m}}$), satisfying Shannon's Theorem [40].

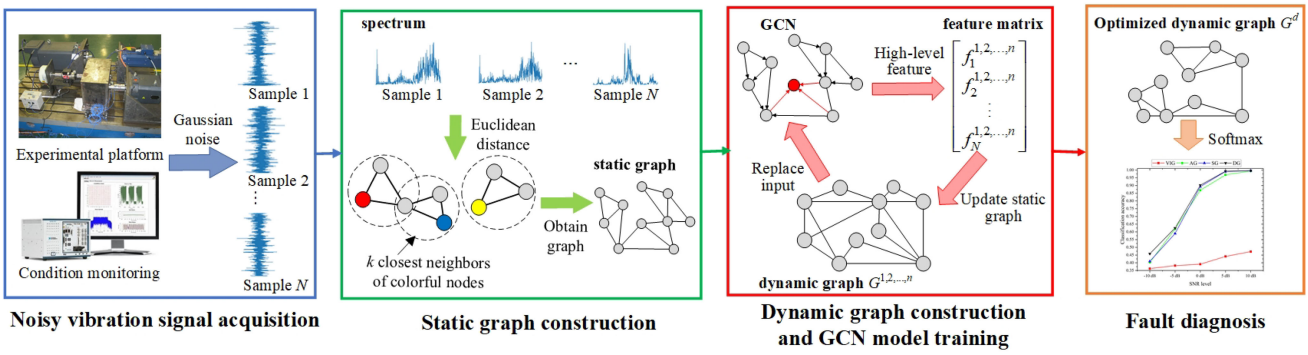


Fig. 5. General process of the proposed diagnosis method.

**Algorithm:**

A. Constructing the SG

**Input:** the monitoring signal $X_1, X_2, \ldots, X_n$, select the distance metric function, select the normalized methods.

**Output:** the SG $G^0$, the feature matrix $F^0$.

1. Normalizing signal: $X_i^{\text{nor}} \leftarrow X_i, i = 1,2,\ldots,n$;
2. Obtaining the sample node representation: $f_i \xleftarrow{\text{FFT}} X_i^{\text{nor}}$;
3. Obtaining the feature matrix: $F^0 \leftarrow f_i$;
4. Calculating the Euclidean distance: $\text{Dis}(f_i, f_j)$;
5. Obtaining the $k$ closest neighbors node set of node $\text{V}_\text{P}$: $\psi(V_p) = \{\tilde{V}_{pi}\}_{i=1}^{k}$, if$\text{Dis}(fV_p, f\tilde{V}_{pi})$is$k$thsamllest;
6. Establishing the edge connections for every node;
7. Output the SG $G^0$ and the feature matrix $F^0$.

B. Constructing the DG and Fault diagnosis

**Input:** the SG $G^0$, the feature matrix $F^0$, epoch $T_1$, $T_2$, $T_3$ ($T_1$<$T_2$<$T_{3)}$, training epoch $M$ for the DG;

**Output:** the fault category $Z$.

1. Dividing the training node set and testing node set: $V_{\text{train}}, V_{\text{test}} \leftarrow V$;
2. Model training:
3. $i = 1$;
4. for $V$ in $V_{\text{train}}$ and $i$<$T_3$ do
5. $Z \leftarrow \text{GCN}(V)$
6. loss
7. updating with backpropagation
8. if ($i$-$T_1$)%M $= 0$ and $i$<$T_2$ do
9. obtaining the high-level feature matrix: $H \leftarrow \text{GCN}(F)$
10. reconstructing the SG: $G^d \leftarrow H$
11. updating the input graph $G^d$
12. recording every DG
13. if $i = \text{T}_2$ do
14. regarding the best DG $G^d$ as the input graph
15. $i = i$+1
16. end for
17. Output the fault category $Z \leftarrow \text{GCN}(V_{\text{test}})$.

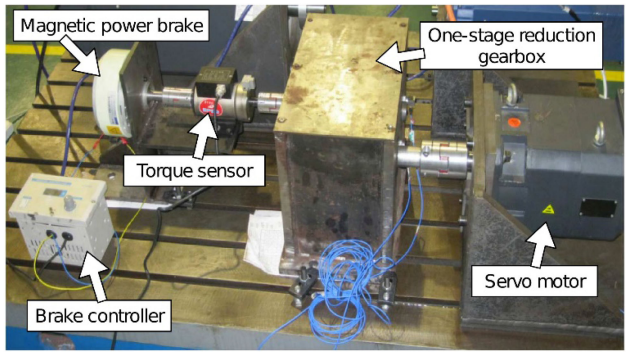


Fig. 6. Practical experimental platform for the one-stage gearbox [19].

TABLE I
SETTING OF THE ONE-STAGE REDUCTION GEARBOX

| Health states | Normal | crack length | | |
|---|---|---|---|---|
| | | 5mm | 10mm | 15mm |
| Load (Nm) | 0/2/4/6/8/10 | | | |
| Shaft speed (rpm) | 1500/900 | | | |
| Samples for training | 72 | 72 | 72 | 72 |
| Samples for testing | 48 | 48 | 48 | 48 |
| Labels | 0 | 1 | 2 | 3 |

On the one-stage reduction gearbox, three kinds of cracked gears were prefabricated. Together with the normal gear, there are four different health states discussed in the experiments. Working conditions with six loads (0, 2, 4, 6, 8, and 10 N·m) and two shaft speeds during gearbox operation are simulated, and samples with the same health state of different loads share the same label. There are 240 samples for each health state with 40 samples for each working condition. Therefore, the number of total samples is 960. There are 1024 data points in each sample. The ratio of the training set and testing set is 3:2, detailed experimental setting is shown in Table I.

***2) Gaussian Noise Addition:*** To verify the performance of processing noisy signals, different signal-to-noise ratios (SNRs) of Gaussian noise, including $-10$, $-5$, 0, 5, and 10 dB, are added

TABLE II
SETTING OF THE ONE-STAGE REDUCTION GEARBOX

| Model | Parameter configuration |
|---|---|
| ChebyNet_1 | Input size: 960*512, structure of GCN: 512-300-100-4, $K$=1. |
| ChebyNet_2 | Input size: 960*512, structure of GCN: 512-300-100-4, $K$=2. |
| ChebyNet_3 | Input size: 960*512, structure of GCN: 512-300-100-4, $K$=3. |
| GCN | Input size: 960*512, structure of GCN: 512-300-100-4. |
| MRF-GCN | Input size: 960*512, structure of GCN: 512-300-100, fully connected layer: 300-4, $K$=1, 2, 3. |

The significance Table II is that our method obtained the best performance.

to the collected signal according to

$$\mathrm{SNR\,(dB)} = 10\log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right) \tag{11}$$

where $P_{\mathrm{noise}}$ is the noise power and $P_{\mathrm{signal}}$ is the signal power.

After adding the Gaussian noise to the collected signal, the SG can be constructed following the steps in Section III. The epochs $T_1$, $T_2$, and $T_3$ were set to 150, 450, and 550, respectively. The training epoch $M$ for the DG was set to 30 according to parameter selection experiments, and the stronger the computing power of the computer, the smaller the value of epochs.

***3) Improvement of the DG:*** Five existing GCN models were used to compare the classification accuracies between the SG and the DG. The five GCN models include a basic GCN [31], three kinds of Chebyshev GCN (ChebyNet) with different kernel $K$ [41], and a multireceptive field GCN (MRF-GCN) [22]. Detailed parameter settings are shown in Table II.

In the model training process, the learning rate was set to 0.01 first, then it changed adaptively. The optimizer was Adam. It is worth mentioning that dropout was added after every graph convolution layer of each model to avoid overfitting. Ten trials were conducted to eliminate the contingency of the results. The average accuracy and time of every experiment are given. The time includes data reading, data preprocessing, graph construction, model training, and testing. Since the time for every SNR-level is approximately similar, an overall time is given, as shown in Table III. The confusion matrixes of the proposed method under five SNRs also are given in Fig. 7 .

From Fig. 7, we can see that the noise energy of the signal gradually decreases, the number of correctly classified samples gradually decreases. It can be seen from Table III that the classification accuracy of the DG is higher than the SG for every GCN model. In addition, the DG-based GCN model is more stable than the SG-based GCN model. Noise will change the $k$ closest neighbors of nodes in the SG so that two nodes with low similarity would be connected. To further improve the quality of the constructed graphs, the DG is constructed using extracted features from GCN. Since the powerful graph features the learning ability of GCN, GCN can mine the fault information from the noisy signals. Therefore, using the output of GCN to reconstruct the relationship between samples may improve the quality of the graph. The GCN model can obtain more effective fault information from the high-quality graph. This is the reason why the DG can reduce the influence of noise.

As the noise energy of the signal gradually decreases, the performance of the SG-based GCN model is almost the same as the DG-based GCN model, but the latter need more time for training. Therefore, when the noise is weak or there is no noise, the diagnosis task can be performed only by constructing the SG.

To be more convincing, the number of total edges and special edges connecting two nodes with the same label are counted. The results are shown in Table IV. Compared to the SG, there are more special edges in the DG, so GCN can aggregate more identical fault features to improve the performance of the DG.

***4) Comparing With Existing Graph Construction Methods:*** The influence of the graph construction method on the GCN model is also investigated. Here, five graph construction methods were adopted, including the vibration-indicator-based graph (VIG) [36], the affinity graph (AG) [22], the SuperGraph [39], the SG, and the DG. The AG was composed of all samples, where 60% of nodes are as training set and 40% are as testing set. The MRF-GCN model was selected for comparison experiments, and the data settings are the same as Table I. The detailed experimental settings for every graph construction method are shown in Table V.

Ten trials were conducted to eliminate the contingency of the results. The average accuracy of every experiment is given, as shown in Fig. 8. The classification accuracies for the DG are best than others, especially when signals are at SNR $= -10$ dB. With the decrease of SNR, the diagnostic accuracy of five graph construction methods decreases. In addition, noise has a great influence on the SuperGraph and VIG, which makes it difficult to characterize the fault information in the SuperGraph and VIG. It also shows that noise will have a great influence on the quality of the graph finally. Therefore, it is necessary to improve the quality of the constructed graph.

***5) Generalization Ability Under Different Loads:*** The generalization ability of the proposed method affects the performance in coping with practical diagnostic tasks, especially under variable working conditions. Therefore, the generalization ability under variable working conditions and SNR levels is investigated. Monitored signals under six different loads (0, 2, 4, 6, 8, and 10 N·m) and two shaft speeds (1500 and 900 r/min) with different SNR levels ($-10$, $-5$, 0, 5, and 10 dB) were used. Verification experiments for generalization ability were conducted, where signals of a certain load were selected as a testing set and others were as the training set. The average results of five trials are listed in Table VI.

Similar classification accuracy in the generalization ability verification experiments can be obtained in Table III. As the SNR level gradually improves, the proposed method can obtain a higher classification accuracy in each testing set, especially when it is more than 0 dB. In addition, the proposed method can achieve satisfactory classification results for each testing set. Therefore, the proposed method has a certain generalization ability.

TABLE III
CLASSIFICATION RESULTS UNDER DIFFERENT SNR LEVELS

| Graph type | GCN model | Average accuracy (%) under different SNR levels | | | | | Time (s) |
|---|---|---|---|---|---|---|---|
| | | -10 dB | -5 dB | 0 dB | 5 dB | 10 dB | |
| Static graph | ChebyNet_1 | 36.33±1.13 | 54.09±2.55 | 78.18±1.61 | 92.37±0.79 | 98.26±0.51 | **39** |
| | ChebyNet_2 | 36.02±1.22 | 52.29±1.81 | 83.59±2.13 | 97.37±0.45 | 98.70±0.23 | 41 |
| | ChebyNet_3 | 33.72±1.05 | 56.33±1.24 | 77.11±2.60 | 97.46±0.16 | 98.52±0.36 | 49 |
| | GCN | 25.02±1.48 | 40.76±1.67 | 70.70±2.88 | 95.39±0.82 | 97.97±0.43 | 43 |
| | MRF-GCN | 38.10±1.20 | 57.84±0.71 | 86.95±0.93 | 98.20±0.14 | 99.24±0.14 | 68 |
| Dynamic graph | ChebyNet_1 | 41.87±0.99 | 50.42±1.67 | 79.97±0.94 | 93.18±1.08 | 98.44±0.37 | 107 |
| | ChebyNet_2 | 42.66±0.84 | 53.14±1.05 | 85.10±0.97 | 97.50±0.51 | 98.66±0.26 | 100 |
| | ChebyNet_3 | 38.86±1.00 | 52.34±1.62 | 75.83±2.04 | 97.99±0.38 | 99.35±0.24 | 103 |
| | GCN | 39.95±1.12 | 53.10±1.04 | 81.98±0.91 | 94.77±0.73 | 98.18±0.26 | 82 |
| | MRF-GCN | **45.89±1.00** | **59.45±0.92** | **87.27±0.84** | **98.39±0.58** | **99.58±0.20** | 121 |

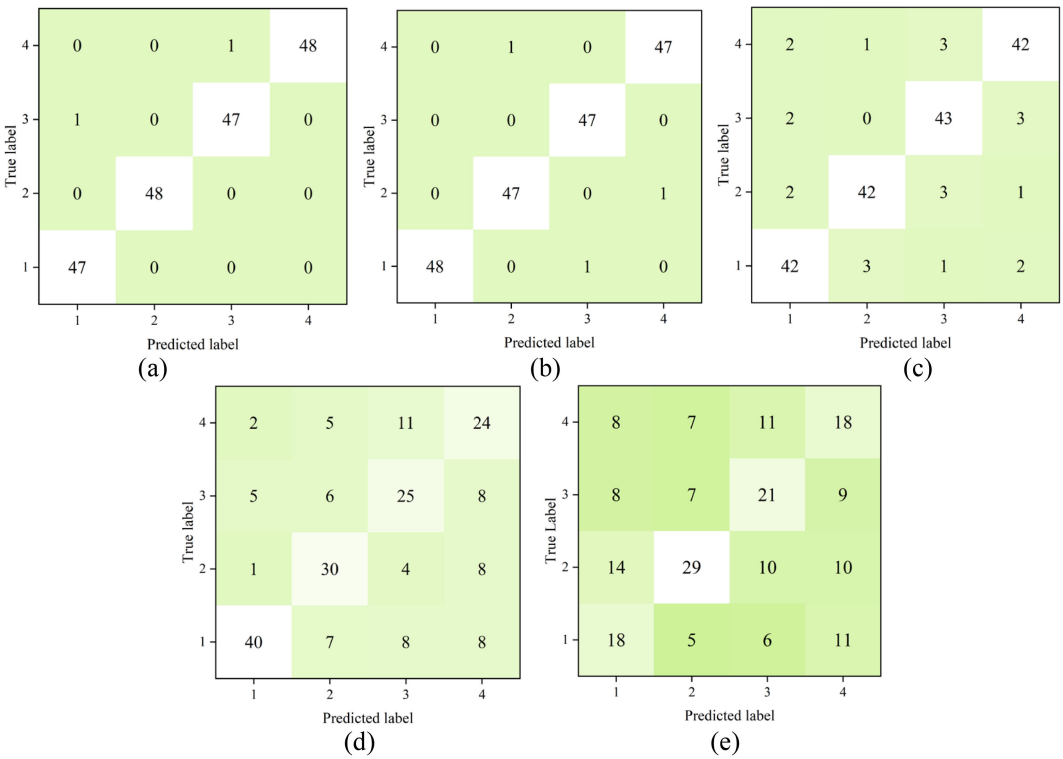


Fig. 7. Confusion matrix of the proposed method under five SNRs. (a) 10 dB. (b) 5 dB. (c) 0 dB. (d) -5 dB. (e) -10 dB.

## B. Planetary Gearbox Platform

***1) Data Description:*** The drivetrain dynamic simulator was used to collect the monitoring data. There are two subdatasets, including gear dataset and bearing dataset, in this platform, and the bearing dataset was used here [42]. Two different working conditions were investigated during experiments, setting the rotating speed system load to be 20 Hz–0V and 30 Hz–2V. There are five health states in the bearing dataset, including four fault types and one normal state. The detailed description of fault types for bearings is shown in Table VII. Therefore, the fault diagnosis of the bearing dataset is a five-class classification task of complex fault types under multiple working conditions. There are 200 samples for each health state, the same number of samples is collected under each working condition. There are 1024 data points in each sample. A total of 1000 samples were obtained in the bearing dataset. 10% of the samples in each health condition were randomly selected as the training set, and the remaining samples were the testing set. The detailed setting of the bearing dataset is shown in Table VIII.

TABLE IV
RESULTS OF EDGE CONNECTIONS FOR TWO GRAPHS

| SNR levels | Number of total edges | | Number of special edges | |
|---|---|---|---|---|
| | Static graph | Dynamic graph | Static graph | Dynamic graph |
| -10 dB | 2210 | 2042 | 248 | 560 |
| -5 dB | 2108 | 1954 | 412 | 686 |
| 0 dB | 1990 | 1864 | 840 | 892 |
| 5 dB | 1682 | 1588 | 958 | 962 |
| 10 dB | 1610 | 1484 | 966 | 970 |

TABLE V
SETTING OF THE GRAPH CONSTRUCTION METHOD

| Method | Parameter configuration |
|---|---|
| VIG [36] | Input size: 960*14, structure of GCN: 14-100-20, fully connected layer: 60-4, $K$=1, 2, 3. |
| AG [22] | $2\beta^2$=1024, $\varepsilon$=0.8, input size: 480*512, structure of GCN: 512-300-100, fully connected layer: 300-4, $K$=1, 2, 3. |
| SuperGraph [39] | Input size: 960*33, structure of GCN: 33-300-100, fully connected layer: 300-4, $K$=1, 2, 3. |
| SG | Input size: 960*512, structure of GCN: 512-300-100, fully connected layer: 300-4, $K$=1, 2, 3. |
| DG | Input size: 960*512, structure of GCN: 512-300-100, fully connected layer: 300-4, $K$=1, 2, 3. |

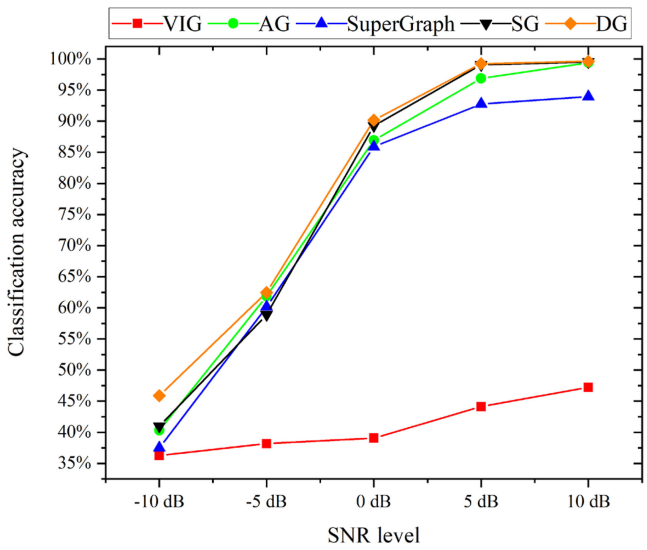


Fig. 8. Classification results of five graph construction methods.

TABLE VI
RESULTS OF GENERALIZATION ABILITY VERIFICATION EXPERIMENTS

| Testing set | Classification accuracy (%) | | | | |
|---|---|---|---|---|---|
| | -10 dB | -5 dB | 0 dB | 5 dB | 10 dB |
| 0 Nm | 38.50±1.22 | 56.50±1.83 | 86.00±1.22 | 96.50±0.93 | 99.25±0.61 |
| 2 Nm | 44.25±1.70 | 66.75±0.61 | 77.25±1.22 | 98.75±1.12 | 99.50±0.61 |
| 4 Nm | 41.50±1.83 | 53.50±0.09 | 75.50±0.61 | 97.50±0.79 | 99.00±0.50 |
| 6 Nm | 50.25±0.94 | 56.50±0.93 | 84.50±0.61 | 95.25±1.22 | 99.25±0.61 |
| 8 Nm | 55.00±1.37 | 61.00±0.50 | 86.75±0.61 | 98.50±0.93 | 99.75±0.40 |
| 10 Nm | 41.50±0.93 | 62.75±0.50 | 80.75±0.61 | 98.00±0.61 | 99.75±0.50 |

TABLE VII
BEARING FAULT TYPES DESCRIPTION

| Fault type | Description |
|---|---|
| Inner | Inner ring fault |
| Ball | Ball fault |
| Outer | Outer ring fault |
| Combination | Mixed with inner ring fault and outer ring fault |

TABLE VIII
SETTING OF THE BEARING DATASET

| Health state | Ball | Inner | Outer | Combination | Healthy |
|---|---|---|---|---|---|
| Traning samples | 10*2 | 10*2 | 10*2 | 10*2 | 10*2 |
| Testing samples | 90*2 | 90*2 | 90*2 | 90*2 | 90*2 |
| Labels | 0 | 1 | 2 | 3 | 4 |

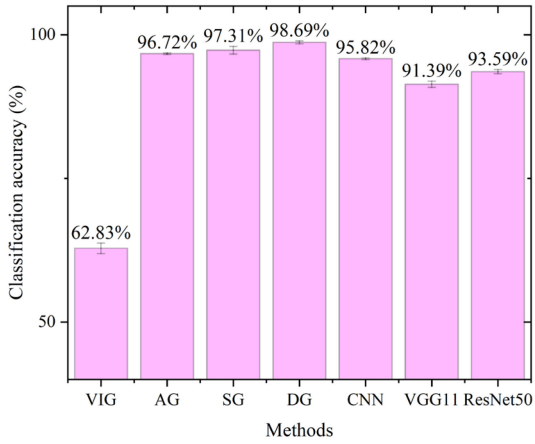


Fig. 9. Classification result of DL-based methods.

***2) Fault Diagnosis Result:*** Traditional DL-based methods often require a large number of training samples to train the model. In order to verify the performance of the proposed method in processing a few training samples, only 10% of the samples were used to train the GCN model. In addition, comparison experiments were conducted to demonstrate the efficiency of the proposed method under a few training samples. Traditional DL-based methods, such as a traditional three-layer CNN, ResNet50, VGG11, VIG, AG, and SG, were considered for comparison. For CNN, VGG11, and ResNet50, the length of the vibration signal is 1024, and after extracting the spectral features, the length of the input feature vector, which is composed of spectral features is 512. After that, the spectral features were reshaped into the size of (16, 16, 2). The structure of GCN in VIG, AG, SG is the same as Table V, only the input size 480 changes to 1000. The results of different DL-based methods are shown in Fig. 9, and the computational complexity of each method is given in Table IX.

It can be seen from Fig. 9 that the classification accuracy of the proposed method is the highest, reaching 98.69%. In addition, the accuracy of those GCN-based DL methods is over 96%. Therefore, by the merits of the powerful graph feature learning

TABLE IX
COMPUTATIONAL COMPLEXITY OF DEEP LEARNING METHODS

| Model | Computational complexity of one layer |
|---|---|
| VIG | $O(K\|E\|)$, $\|E\|$=20000. |
| AG | $O(K\|E\|)$, $\|E\|$=1908. |
| SG | $O(K\|E\|)$, $\|E\|$=1732. |
| DG | $O(K\|E\|)$, $\|E\|$=1690. |
| CNN | $O(M^2K_e^2C_{in}C_{out})$ |
| VGG11 | $O(M^2K_e^2C_{in}C_{out})$ |
| ResNet50 | $O(M^2K_e^2C_{in}C_{out})$ |

Noted: $M$ is the size of the output feature map, $K_e$ is the size of the convolution kernel, $|E|$ is the number of edges, $C_{in}$ is the number of input channels, and $C_{out}$ is the number of output channels.

ability of GCN, the proposed method can process the diagnosis task using a few training samples. As shown in Table VIII, among the four graphs, the number of edges in the DG is the least, which means that MRF-GCN can learn more fault information with a small number of edges on the DG. The DG reduces the computational complexity of graph feature extraction by reducing redundant edges.

## VI. CONCLUSION

In this article, a DG-based feature learning with few edges considering noisy samples was proposed for the rotating machinery fault diagnosis. The influence of noise on the signal-feature-based graph was explored. The proposed DG not only reduced the redundant edges but also improved the quality of the noisy-signal-feature-based graph. In addition, verification experiments were conducted on two practical platforms, and the results verified the diagnosis performance of the proposed method. Compared to the SG, the DG can improve the diagnosis performance of commonly used GCN models. The DG-based GCN model has the ability to learn the feature representations of noisy signals. For the noise-free signals, GCN can achieve excellent diagnosis results using limited training samples by learning graph features from the DG that has few edges. The proposed DG-based feature learning method has the potential to be applied for the online fault diagnosis.

However, only the node information was considered in the construction of a DG, ignoring the edge information. For a node, the priority of the edge connections between its $k$ closest neighbor should be higher than the other edges. In order to further improve the quality of the constructed graph, the weight on the edge should be assigned and investigated in the future work.

## REFERENCES


[1] G. Xu, M. Liu, Z. Jiang, W. Shen, and C. Huang, "Online fault diagnosis method based on transfer convolutional neural networks," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 509–520, Feb. 2020.

[2] M. F. Tariq, A. Q. Khan, M. Abid, and G. Mustafa, "Data-driven robust fault detection and isolation of three-phase induction motor," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4707–4715, Jun. 2019.

[3] Z. Hu, Y. Wang, M. F. Ge, and J. Liu, "Data-driven fault diagnosis method based on compressed sensing and improved multi-scale network," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3216–3225, Apr. 2020.

[4] D. Goyal and B. S. Pabla, "The vibration monitoring methods and signal processing techniques for structural health monitoring: A review," *Arch. Comput. Method Eng.*, vol. 23, no. 4, pp. 585–594, Dec. 2016.

[5] M. Cerrada *et al.*, "A review on data-driven fault severity assessment in rolling bearings," *Mech. Syst. Signal Process.*, vol. 99, pp. 169–196, Jan. 2018.

[6] P. Liu, H. Li, and P. Ye, "A method for rolling bearing fault diagnosis based on sensitive feature selection and nonlinear feature fusion," in *Proc. 8th Int. Conf. Intell. Computation Technol. Automat.*, 2015, pp. 30–35.

[7] Q. Fu, B. Jing, P. He, S. Si, and Y. Wang, "Fault feature selection and diagnosis of rolling bearings based on EEMD and optimized Elman_adaboost algorithm," *IEEE Sensors J.*, vol. 18, no. 12, pp. 5024–5034, Jun. 2018.

[8] A. Hajnayeb, A. Ghasemloonia, S. E. Khadem, and M. H. Moradi, "Application and comparison of an ANN-based feature selection method and the genetic algorithm in gearbox fault diagnosis," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10205–10209, Aug. 2011.

[9] R. Patel and V. K. Giri, "Feature selection and classification of mechanical fault of an induction motor using random forest classifier," *Perspectives Sci.*, vol. 8, pp. 334–337, 2016.

[10] K. Han, Y. Wang, C. Zhang, C. Li, and C. Xu, "Autoencoder inspired unsupervised feature selection," in *IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pp. 2941–2945, 2018.

[11] X. Li *et al.*, "Intelligent cross-machine fault diagnosis approach with deep auto-encoder and domain adaptation," *Neurocomputing*, vol. 383, no. 28, pp. 235–247, Mar. 2020.

[12] Z. He, H. Shao, X. Zhang, J. Cheng, and Y. Yang, "Improved deep transfer auto-encoder for fault diagnosis of gearbox under variable working conditions with small training samples," *IEEE Access*, vol. 7, pp. 115368–115377, Aug. 2019.

[13] Z. Chen, A. Mauricio, W. Li, and K. Gryllias, "A deep learning method for bearing fault diagnosis based on cyclic spectral coherence and convolutional neural networks," *Mech. Syst. Signal Process.*, vol. 140, Jun. 2020, Art. no. 106683.

[14] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, and M. Gabbouj, "Real-time fault detection and identification for MMC using 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8760–8771, Nov. 2019.

[15] H. Shao, M. Xia, G. Han, Y. Zhang, and J. Wan, "Intelligent fault diagnosis of rotor-bearing system under varying working conditions with modified transfer convolutional neural network and thermal images," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3488–3496, May 2021.

[16] M. Qiao, S. Yan, X. Tang, and C. Xu, "Deep convolutional and LSTM recurrent neural networks for rolling bearing fault diagnosis under strong noises and variable loads," *IEEE Access*, vol. 8, pp. 66257–66269, 2020.

[17] Y. Han, W. Qi, N. Ding, and Z. Geng, "Short-time wavelet entropy integrating improved LSTM for fault diagnosis of modular multilevel converter," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2020.3041850.

[18] S. Ye, J. Jiang, J. Li, Y. Liu, Z. Zhou, and C. Liu, "Fault diagnosis and tolerance control of five-level nested NPP converter using wavelet packet and LSTM," *IEEE Trans. Power Electron.*, vol. 35, no. 2, pp. 1907–1921, Feb. 2020.

[19] J. Liu, Y. Hu, Y. Wang, B. Wu, J. Fan, and Z. Hu, "An integrated multi-sensor fusion-based deep feature learning approach for rotating machinery diagnosis," *Meas. Sci. Technol.*, vol. 29, no. 5, 2018, Art. no. 055103.

[20] L. Wen, L. Gao, X. Li, and B. Zeng, "Convolutional neural network with automatic learning rate scheduler for fault classification," *IEEE Trans. Instrum. Meas.*, vol. 70, Jan. 2021, Art. no. 3509912, doi: 10.1109/TIM.2020.3048792.

[21] M. Ma and Z. Mao, "Rotating machinery prognostics via the fusion of particle filter and deep learning," *Struct. Health Monit.*, 2019, doi: 10.12783/shm2019/32300.

[22] T. Li, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Multi-receptive field graph convolutional networks for machine fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 12739–12749, Dec. 2021.

[23] T. Wang, G. Lu, and P. Yan, "A novel statistical time-frequency analysis for rotating machine condition monitoring," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 531–541, Jan. 2020.

[24] C. H. Lo, Y. K. Wong, A. B. Rad, and K. M. Chow, "Fusion of qualitative bond graph and genetic algorithms: A fault diagnosis application," *ISA Trans.*, vol. 41, no. 4, pp. 445–456, Oct. 2002.

[25] Y. Gao and D. Yu, "Total variation on horizontal visibility graph and its application to rolling bearing fault diagnosis," *Mechanism Mach. Theory*, vol. 147, May 2020, Art. no. 103768.
[26] W. Nie, M. Ren, A. Liu, Z. Mao, and J. Nie, "M-GCN: Multi-branch graph convolution network for 2D image-based on 3D model retrieval," *IEEE Trans. Multimedia*, vol. 23, pp. 1962–1976, Jul. 2021.
[27] K. Guo *et al.*, "Optimized graph convolution recurrent neural network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1138–1149, Feb. 2021.
[28] X. He *et al.*, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.
[29] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," 2017, *arXiv:1612.04844v2*.
[30] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. 33rd AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 7370–7377, 2019.
[31] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2017.
[32] D. Zhang, E. Stewart, M. Entezami, C. Roberts, and D. Yu, "Intelligent acoustic-based fault diagnosis of roller bearings using a deep graph convolutional network," *Measurement*, vol. 156, May 2020, Art. no. 107585.
[33] C. Li, L. Mo, and R. Yan, "Rolling bearing fault diagnosis based on horizontal visibility graph and graph neural networks," in *Proc. Int. Conf. Sens., Meas. Data Analytics Era Artif. Intell.*, 2020, pp. 275–279, doi: 10.1109/ICSMD50554.2020.9261687.
[34] X. Yu, B. Tang, and K. Zhang, "Fault diagnosis of wind turbine gearbox using a novel method of fast deep graph convolutional networks," *IEEE Trans. Instrum. Meas.*, vol. 70, Jan. 2021, Art no. 6502714, doi: 10.1109/TIM.2020.3048799.
[35] T. Li, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction," *Rel. Eng. Syst. Saf.*, vol. 215, 2021, Art. no. 107878, doi: 10.1016/j.ress.2021.107878.
[36] S. Wang, S. Xing, Y. Lei, N. Lu, and N. Li, "Vibration indicator-based graph convolutional network for semi-supervised bearing fault diagnosis," in *Proc. IOP Conf. Series: Mater. Sci. Eng*, vol. 1043, p. 052026, 2020.
[37] R. Paredes and E. Chávez, "Using the k-nearest neighbor graph for proximity searching in metric spaces," in *Proc. String Process. Inf. Retrieval*, 2005, pp. 127–138.
[38] K. Hajebi, Y. Abbasi-Yadkori, H. Shahbazi, and H. Zhang, "Fast approximate nearest-neighbor search with k-nearest neighbor graph," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1312–1317.
[39] C. Yang, K. Zhou, and J. Liu, "SuperGraph: Spatial-temporal graph-based feature extraction for rotating machinery diagnosis," *IEEE Trans. Ind. Electron.*, to be published, doi: 10.1109/TIE.2021.3075871.
[40] Y. Hu, J. Fan, and Y. Jin, "Study of the influences of transient crack propagation in a pinion on time-varying mesh stiffness," *Shock Vib.*, vol. 6, 2016, Art. no. 6928686.
[41] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016.
[42] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly accurate machine fault diagnosis using deep transfer learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.

**Kaibo Zhou** received the B.E., M.S., and Ph.D. degrees in control engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991, 1995, and 1998, respectively.

He is currently a Professor with the School of Artificial Intelligence and Automation, HUST. His research interests include condition monitoring and fault diagnosis, image processing, and deep learning.

**Chaoying Yang** received the B.S. degree from North China Electric Power University, Baoding, China, in 2020. He is currently working toward the M.S. degree in control engineering with the Huazhong University of Science and Technology (HUST), Wuhan, China.

His research interests include graph theory, fault diagnosis, and deep learning.

**Jie Liu** (Member, IEEE) received the B.E. and M.S. degrees in control engineering and the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2010, 2013, and 2018, respectively.

He is currently an Assistant Professor with the School of Civil and Hydraulic Engineering, HUST. Prior to joining the HUST in 2018, he was a Visiting Student with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests include machine health condition monitoring and prognostics.

**Qi Xu** received the B.S. and M.S. degrees in control engineering and the Ph.D. degree in water conservancy and hydropower engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1992, 1995, and 2001, respectively.

She has been an Associate Professor with the School of Artificial Intelligence and Automation, HUST, since 2007. Her research interests include functional electrical stimulation, wireless power transfer, and implantable devices.