Applied Intelligence (2023) 53:21211–21226
https://doi.org/10.1007/s10489-023-04665-7

Rotating machinery fault diagnosis based on feature extraction via an unsupervised graph neural network

Jing Feng[1,2] · **Shouyang Bao**[1,2] · **Xiaobin Xu**[1,2] · **Zhenjie Zhang**[1,2] · **Pingzhi Hou**[1,2] · **Felix Steyskal**[3] · **Schahram Dustdar**[4]

Accepted: 21 April 2023 / Published online: 23 May 2023

Abstract

Fault diagnosis is an essential process for the health maintenance of rotating machinery. With the development of AI technology, many deep learning-based methods have been applied to fault diagnosis to enhance the intelligence level of equipment maintenance. Such methods normally need a large amount of labeled data for model training. However, label acquisition is a difficult task that requires extensive human labor. To address these issues, a fault diagnosis method based on feature extraction via an unsupervised graph neural network is proposed in this paper. In the proposed method, the K-nearest neighbor approach is adopted to construct a fault graph from the collected signals, thereby providing extra relationship information for fine feature mining. Then, the GraphSAGE model is trained on the constructed graph in an unsupervised way, that is, it does not need labeled data, to extract features of each signal sample. Based on the extracted features, some traditional classifiers are adopted to identify the fault types. The proposed model is evaluated on a rolling bearing dataset provided by the University of Paderborn and a motor rotor dataset collected by a constructed motor rotor system. Compared with some traditional deep learning-based fault diagnosis methods, the proposed model can achieve more accurate diagnoses even when there are only a few labeled samples.

Keywords Fault Diagnosis · Graph Neural Network · Unsupervised GraphSAGE · Classification

1 Introduction

Rotating parts in power machinery and working machinery easily experience faults after long-term operation. It is essential to detect failures early, thereby preventing deadly accidents and saving maintenance costs [1]. Therefore, fault diagnosis plays a vital role in industrial manufacturing [2]. The aim of fault diagnosis is to identify the type of fault, the location where the fault occurred, or even the extent of the fault in the running device. In the early stage, fault detection relied heavily on manual inspection, which costs considerable manpower. With the development of sensor technology, various sensors, e.g., acceleration sensors, speed sensors and displacement sensors, can be installed to monitor the working states of devices. The distribution and amplitude as well as the impulses on the waveform of signals that are collected from the sensors vary with different fault conditions; thus, signals are able to reflect the operating status of a device.

Intuitively, signal analysis is deployed to explore distinguished signal features for fault diagnosis. For example, time domain analysis [3], frequency domain analysis [4] and time–frequency analysis [5] are adopted to extract signal features by analyzing signal properties. In such methods, the effect of extracted features highly relies on signal quality. Noise is unavoidable in the fault diagnosis stage. Moreover, entropy-based indicators [6], such as sample entropy (SE), permutation entropy (PE) and fuzzy entropy (FE), are widely applied to extract signal features by quantifying the amount of randomness in a signal. The choice of the indicators has a high impact on the performance of feature extraction. Additionally, with the development of sensor technology, the

✉ Xiaobin Xu
xuxiaobin1980@163.com

1 School of Automation, Hangzhou Dianzi University, Hangzhou, Zhejiang, China

2 China-Austria Belt and Road Joint Laboratory On Artificial Intelligence and Advanced Manufacturing, Hangzhou Dianzi University, Hangzhou, Zhejiang, China

3 M-U-T GmbH, Stockerau, Austria

4 Distributed Systems Group, TU Vienna, Vienna, Austria

amount and variety of signals are augmented, and extracting features by the above methods may be time-consuming or produce redundant information.

In recent years, deep learning-based algorithms, such as CNNs [7] and RNNs [8], have been increasingly applied in fault diagnosis. Such methods integrate feature extraction with classification, thereby avoiding the selection of features. Nonetheless, a large number of labeled samples are required to train such deep models to realize their advantages in mining massive fault data. However, it is usually difficult to obtain a large number of labeled samples in actual working conditions. Training a deep neural network in a label scarcity situation cannot achieve a good convergence state, which leads to insufficient network model expressions and poor generalization performance. Therefore, some deep representation learning methods [9, 10], such as autoencoders (AEs), are capable of extracting features and establishing complex mapping relationships between fault features and fault categories. AEs depend less on prior knowledge. Although such deep representation learning methods can achieve feature extraction without the aid of labeled data, they consider only each signal sample and not the relationship between the samples, which may cause insufficient mining outcomes.

Moreover, a graph is a kind of structured data that consists of a set of nodes and edges. Various kinds of data can be modeled as a graph to convey the relationships between data samples. Moreover, a graph neural network (GNN) [11, 12], which is a deep learning method based on graph domain analysis, has been proven to be an excellent model for mining knowledge from graphs. Specifically, graph embedding learning, which aims to represent nodes in a low-dimensional vector space by preserving the topological structure of the graph and node content information, has been proposed so that graph analysis tasks, such as node classification, link prediction and graph clustering, can be easily used in some traditional machine learning algorithms, such as logistic regression, support vector machine classification and K-means clustering. Through a graph structure, graph embedding learning can mine information from nodes in addition to node relationship information, and this information can be used to obtain plentiful features. Furthermore, the interactive mechanism aids graph embedding learning to achieve better feature extraction by sharing information.

In this paper, to overcome the drawbacks of signal-based methods and deep learning-based methods, which may suffer from insufficient feature extraction, we apply graph embedding learning to fault diagnosis scenarios for fine feature extraction, thereby achieving accurate fault diagnosis. Specifically, to explore the relationship among the fault samples, a fault graph is constructed from the collected vibration signals to convey the extra features for fault samples. Moreover, a graph sample and aggregate (GraphSAGE) method [13] is adopted in an unsupervised way to further extract features from the constructed fault graph. Finally, some traditional machine learning-based classifiers are implemented on the extracted features to identify the faulty statuses of the devices. The proposed method is implemented on two rotating machinery datasets for further evaluation.

In summary, a fault diagnosis method based on feature extraction via an unsupervised graph neural network is proposed in this paper. The contributions of the proposed method are listed as follows:

1. In this paper, the proposed fault diagnosis method models a signal into a graph that considers the information of samples and the relationship between them; thus, more information can be mined from the signals.
2. The proposed fault diagnosis method trains a supervised model in an unsupervised way that does not need the aid of labeled data, and the learned features are more comprehensive and can be combined with many machine learning methods.
3. The proposed fault diagnosis method is able to achieve an accurate diagnosis even when only a few labeled samples are available.

The rest of the paper is structured as follows: Section 2 summarizes the related work of the proposed method. Section 3 introduces the theoretical principle of GraphSAGE. Section 4 demonstrates a feature extraction method based on unsupervised GraphSAGE. Section 5 details the procedure of the proposed fault diagnosis method. Section 6 presents the results and discusses two cases. Finally, a conclusion of the paper is given in Section 7.

2 Related work

2.1 Signal analysis-based feature extraction in fault diagnosis

In fault diagnosis, feature extraction is a key process responsible for extracting relevant information from raw signals. Signal analysis is a typical method for extracting features from signals and has been used for decades. Specifically, time domain features, such as the standard deviation, skewness, and root mean square (RMS) value, and frequency domain features, such as the spectral mean, spectral crest factor, spectral entropy, roll-off and energies, can be obtained to describe the characteristics of the signals. Furthermore, time–frequency analysis, which is able to represent the energy or strength of signals in both time and frequency, is adopted for feature extraction. For example, Chen et al. [14] proposed a rolling bearing fault diagnosis method in which wavelet thresholding denoising is used to reduce noise in vibration signals, and the method named CEEMDAN which is complete ensemble empirical mode decomposition with adaptive noise energy entropy, is used to extract features from the denoised signals. Finally, features are input into the particle swarm optimization least squares support vector machine (PSO-LSSVM) to achieve fault identification. Boztas and Tuncer [15] combined the discrete wavelet transform (DWT) with an improved one-dimensional local binary method (DC-1D-LABP) to extract signal features, which were then fed into traditional machine learning models for fault diagnosis. Although the improved one-dimensional local binary method (DC-1D-LABP) has the advantage of low computational complexity, it only considers local information, which is one-sided. Overall, signal analysis-based methods are often unable to extract signal features, and they are normally treated as preprocessing methods. In addition, in the face of various types of signal analysis methods, the result of fault diagnosis is highly related to the feature selection phase.

2.2 Entropy-based feature extraction in fault diagnosis

Moreover, entropy-based feature extraction is widespread in fault diagnosis. Compared with signal-based methods, entropy-based methods are statistical calculations on signals; thus, they are robust to noise. Many papers have been published in this field. For example, multiscale entropy is widely adopted in extracting fault features. Zhang et al. [16] proposed a two-stage fault diagnosis method that feeds multiscale entropy features to an optimized support vector machine (SVM). But when there are lost in a signal, multiscale entropy has a poor stability of the fault feature. To solve the problem, some improved entropy-based methods, such as multi-scale permutation entropy [17, 18] and hierarchical multiscale permutation entropy [19], have proposed to improve the stability of fault feature extraction. Except for multiscale entropy, there are many other entropy-based indicators. For example, Leite et al. [20] discussed the performance of the 12 entropy-based features for the monitoring and detection of bearing faults, such as power spectrum entropy (PSE), permutation entropy (PE), wavelet entropy (WE). Therefore, the scale of such intuitive feature indicators is large, and deciding which kind of feature should be chosen is a difficult task. Moreover, such methods are unable to fully mine information from signals.

2.3 Deep learning-based feature extraction in fault diagnosis

Furthermore, a deep learning method can be adopted to learn features from data. Compared with the two types of methods mentioned above, deep learning-based methods can capture more complex and high-level features. For example, Liu et al. [21] proposed a motor fault diagnosis method by combining a deep sparse network (DSF) with improved logistic regression. However, deep sparse network is hard to train due to its sparse connection. Then autoencoder and its improved models are widely applied to fault feature extraction. For example, Li et al. [22] integrated an autoencoder structure with K-means clustering to develop an unsupervised fault diagnosis of rotating machinery. Xiao et al. [23] integrated a modified stacking denoising autoencoder (SDA) with a transfer component analysis (TCA)-based dimensionality reduction method to build a diagnostic model. Xia et al. [24] learned representative features of faulty data by applying a denoising autoencoder in an unsupervised manner. Luo et al. [25] introduced transfer learning into an improved stacked autoencoder based on convolutional shortcuts and a domain fusion strategy to achieve domain feature fusion for rolling bearing fault diagnosis. In this case, fault sample information may not be mined adequately. Although such deep representation learning methods can achieve feature extraction without the aid of labeled data, they consider only each signal sample and not the complex relationship between the samples, which may cause insufficient mining outcomes.

2.4 Graph neural network-based fault diagnosis

Typical graph neural network algorithms, such as the graph convolutional neural network (GCN) [11], GraphSAGE [13], and GAT [26], have already achieved outstanding performance in many fields. In particular, GraphSAGE has been proven to be a powerful tool for node embedding learning. In the method, node embedding is learned by sampling and aggregating the features of its neighboring nodes. In recent years, an increasing number of researchers have paid attention to applying graph neural networks to fault diagnosis [27]. For example, Gao et al. [28] proposed a rotating machinery fault diagnosis method based on a GCN. In this paper, a graph is constructed by implementing KNN on the FFT feature of the signals. Then, a semisupervised graph convolutional neural network is adopted to identify the fault status of each sample. Kavianpour et al. [29] proposed an approach that introduces ARMA to a GCN for structure information extraction. Combined with multilayer multikernel local maximum mean discrepancy, a method is proposed to solve the missing data and changing operation condition problem in fault diagnosis. However, compared with the GCN, GraphSAGE is more scalable. When acquiring the embedding of a new node, a GCN needs all nodes to participate in training, but GraphSAGE only needs to obtain a subgraph for each node by the sampling technique. Therefore, we proposed a fault diagnosis method based on GraphSAGE in this paper.

3 Basic theoretical knowledge of GraphSAGE

The graph sample and aggregate (GraphSAGE) framework is an inductive learning framework for the generation of node embedding by adopting the attribute information of nodes.

The key idea of GraphSAGE is to generate a node embedding by learning a function of aggregating the representation of its neighbor nodes. As the name implies, neighbor sampling and feature aggregation are the two most important steps of the method. Figure 1 demonstrates the neighbor sampling and feature aggregation steps of GraphSAGE, and the details of the two phases are illustrated as follows:

Neighbor sampling Neighbors are randomly sampled with a fixed size layer by layer. Taking the embedding calculation of a node s_i (the circle node shown in Fig. 1) as an example, the first-order neighbors $N^1(s_i)$ (the square nodes) and the second-order neighbors $N^2(s_i)$ (the triangle nodes) are considered with the sampling scale, which is set to $k = 10$ and $k = 5$, respectively, in this paper. Then, the total number of first-order neighbors $N^1(s_i)$ sampled cannot exceed 10. The total number of second-order neighbors $N^2(s_i)$ sampled cannot exceed $10 \times 5 = 50$. Therefore, the circle node s_i can be expressed by a two-layer subgraph (the combination of the circle node, square nodes and triangle nodes) with a limited node number, which is able to save computational costs.

Feature aggregation When a subgraph is obtained, the aggregation operation propagates in the opposite direction of neighbor sample, which is from the second-order neighbors $N^2(s_i)$ to the first-order neighbors $N^1(s_i)$ and then to the target node s_i. To be specific, Eqs. (1–4) illustrate the progress of feature aggregation. Suppose the target node s_i stands for any sample, n_1 is any first-order neighbor of s_i and n_2 is any first-order neighbor of n_1. First, the representation vector of any first-order neighbor z_{n_1} can be calculated by Eq. (1) and Eq. (2). In the equation, mean aggregation is adopted to acquire h_{n_1}, which is the average feature of n_2 corresponding to n_1. Then, a nonlinear transformation (ReLU) is performed to generate the representation of the first-order neighbors z_{n_1} by concatenating the feature of any first-order neighbor node n_1 and its corresponding mean aggregation feature value h_{n_1}. In the same way, embedding of the target node s_i, set as z_{s_i}, can be calculated by Eq. (3) and Eq. (4). When the representation of any first-order neighbors z_{n_1} is obtained, the mean aggregation feature h_{s_i} can be acquired, and then the embedding of the target node z_{s_i} can be calculated by concatenating the feature of s_i with its first-order neighbor representation h_{s_i}.

$$h_{n_1} = mean\left(\left\{\text{feature}(n_2), n_2 \in N^1(n_1)\right\}\right), n_1 \in N^1(s_i) \quad (1)$$

$$z_{n_1} = \mathrm{Re}\,LU\left(W_2 \cdot concat\left(h_{n_1}, \text{feature}(n_1)\right)\right), n_1 \in N^1(s_i) \quad (2)$$

$$h_{s_i} = mean\left(z_{n_1}, n_1 \in N^1(s_i)\right) \quad (3)$$

$$z_{s_i} = \mathrm{Re}\,LU\left(W_1 \cdot concat\left(h_{s_i}, \text{feature}(s_i)\right)\right) \quad (4)$$

In summary, as illustrated above, the embedding of node s_i can be learned by aggregating the node attributes based on the graph structure, which integrates both feature information and the structure information.

4 Feature extraction based on unsupervised GraphSAGE

In the fault diagnosis scenario, sensors are installed to monitor the running states of rotating machinery. In different states, signals are collected from sensors. The variation in the signals reflects the different states of the device. Thus, fault features can be extracted by mining the signals. In this paper, a feature extraction method based on unsupervised GraphSAGE is proposed for fault diagnosis. First, a fault graph is constructed from fault samples by K-nearest neighbor (KNN) [30]. Then, an unsupervised GraphSAGE model is utilized to learn node embeddings from the fault graph

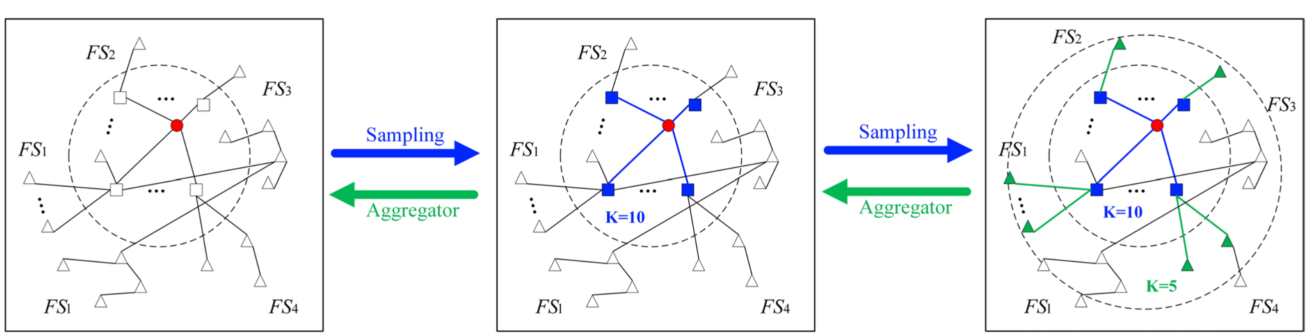


Fig. 1 The sampling and aggregation process of GraphSAGE

to achieve feature extraction of the signals. Finally, some traditional classifiers are implemented on the learned feature to identify various faulty states. The details of each step are shown below.

4.1 Constructing a graph from signals based on K-nearest neighbor

In the fault diagnosis case, uniform sampling is continuously implemented on signals to generate samples of each fault state. As nodes and edges are two important elements in a graph, signal samples are set as nodes, and the similarities between the samples are set as edges. In this paper, K-nearest neighbor is adopted to construct a graph from the signal samples. Figure 2 demonstrates the progress of constructing a fault graph of l fault states $FS = \{FS_1, FS_2, \ldots, FS_l\}$, and $s = \{s_1, s_2, \ldots, s_n\}$, where n is the number of fault samples and stands for all signal samples. As shown in the figure, the fast Fourier transform (FFT) is used to analyze the frequency domain characteristics of each fault sample. Then, K-nearest neighbor (KNN) is adopted to construct a fault graph $G(V, E)$, with fault samples s set as nodes V, frequency domain characteristics $FFT(s)$ set as node attributes and the similarity between any fault sample pair $Sim(s_i, s_j)$ set as the edge E, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$.

For every fault sample s_i, the Euclidean distance between the sample and any other fault sample s_j is calculated by Eq. (5).

$$Sim(s_i, s_j) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(FFT(s_i) - FFT(s_j)\right)^2} \tag{5}$$

Then, fault samples s_j are ranked by similarity, and the top K fault samples are selected as neighbors of s_i. In other words, edges between s_i and the top K fault samples are constructed. By analogy, every fault sample is connected with its K nearest neighbors to form a fault graph. As shown in Fig. 2, the constructed fault graph is able to convey not only the frequency feature of fault samples but also the relationship between the fault samples. The relationship is able to aid the method in better understanding the fault samples; thus, more accurate fault identification can be achieved.

4.2 Learning node embedding by training unsupervised GraphSAGE

In this paper, GraphSAGE is applied to the constructed fault graph to learn the features of each signal sample. Based on the node embedding learning process illustrated in Sect. 3, the feature of a signal sample can be learned by aggregating the FFT feature of its neighbor samples layer by layer,

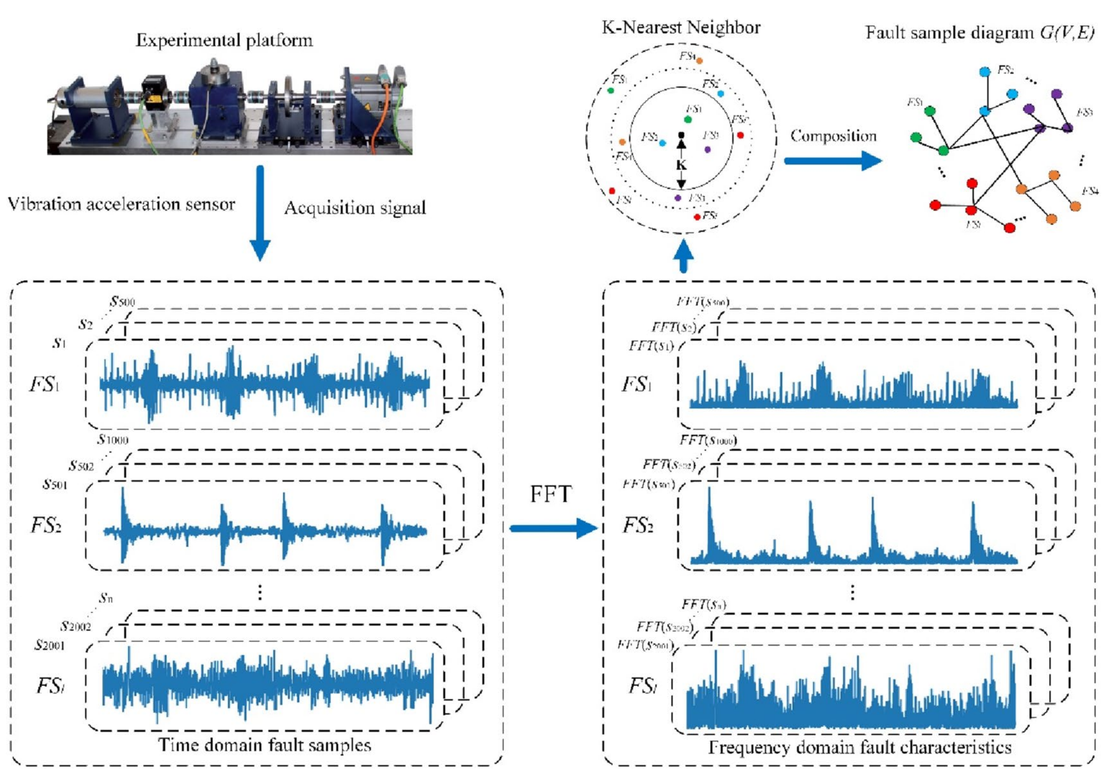


Fig. 2 Constructing a fault graph from the signals by KNN

which combines both node frequency feature information and structure information. As shown in Eq. (2) and Eq. (4), in the established GraphSAGE network, W_1 and W_2 are the network parameters that need to be trained to obtain a better node representation. Thus, labeled samples are normally necessary to aid the training process. However, the acquisition of the labeled sample is difficult and costly in fault diagnosis scenarios. Therefore, GraphSAGE is trained in an unsupervised manner. Figure 3 demonstrates the progress of learning node embedding by unsupervised GraphSAGE.

As shown in the figure, a node pairwise prediction model is constructed to provide the labels for training the GraphSAGE model. First, uniform random walks are implemented on the fault graph to generate a random walk. Then, node pairs are selected from these random walks and are marked with positive labels. Then, node pairs without connections that are considered to have negative labels are randomly sampled from the fault graph with equal numbers of positive ones. Next, embeddings of those selected node pairs can be acquired by concatenating the node embeddings that can be learned by GraphSAGE. Finally, the embeddings of the selected positive and negative node pairs are fed into a dense layer for node pair label prediction.

In summary, an unsupervised GraphSAGE model is mainly constructed by a two-layer GraphSAGE and node pair classification model. Specifically, GraphSAGE layers provide a mechanism to learn the node embeddings, and the node pair classification model intuitively adopts the information of edge existence to construct the training process of the unsupervised model. In the model, binary cross entropy is adopted as the loss function, as shown in Eq. (6), where N is the number of selected node pairs, y_i is the positive label and $p(y_i)$ is the probability of positive affinity, as shown in Fig. 3. During training, the Adam operator is adopted to optimize the model.

$$Loss = -1/N\sum_{i=1}^{N} y_i log\left(\sigma\left(p\left(y_i\right)\right)\right) + \left(1 - y_i\right) log\left(\sigma\left(1 - p\left(y_i\right)\right)\right) \tag{6}$$

In this paper, with the above mechanism, after training for several epochs, 30 is chosen, and then features of fault samples are extracted. Several important hyperparameters are set as follows: the numbers of sampled neighbors in the two layers of GraphSAGE are set as 10 and 5, the number of hidden nodes in the classification layer is set as 128, the learning rate is set as 0.001, the walk length of the random walk is set as 100, and the batch size is set as 50.

5 Fault diagnosis based on feature extraction via an unsupervised GraphSAGE model

By implementing unsupervised GraphSAGE on the constructed fault graph, the features of fault samples are obtained. Based on the features, several traditional classifiers, such as logistic regression, decision tree, random forest, SVM, MLP, KNN, Gaussian NB, and AdaBoost, are implemented on the extracted features to identify the fault type. Table 1 presents the general introduction and parameter settings of these classifiers in this paper.

In conclusion, the pipeline of fault diagnosis based on feature extraction via an unsupervised GraphSAGE model is shown in Fig. 4. The procedure of the proposed fault diagnosis method is shown in detail as follows.

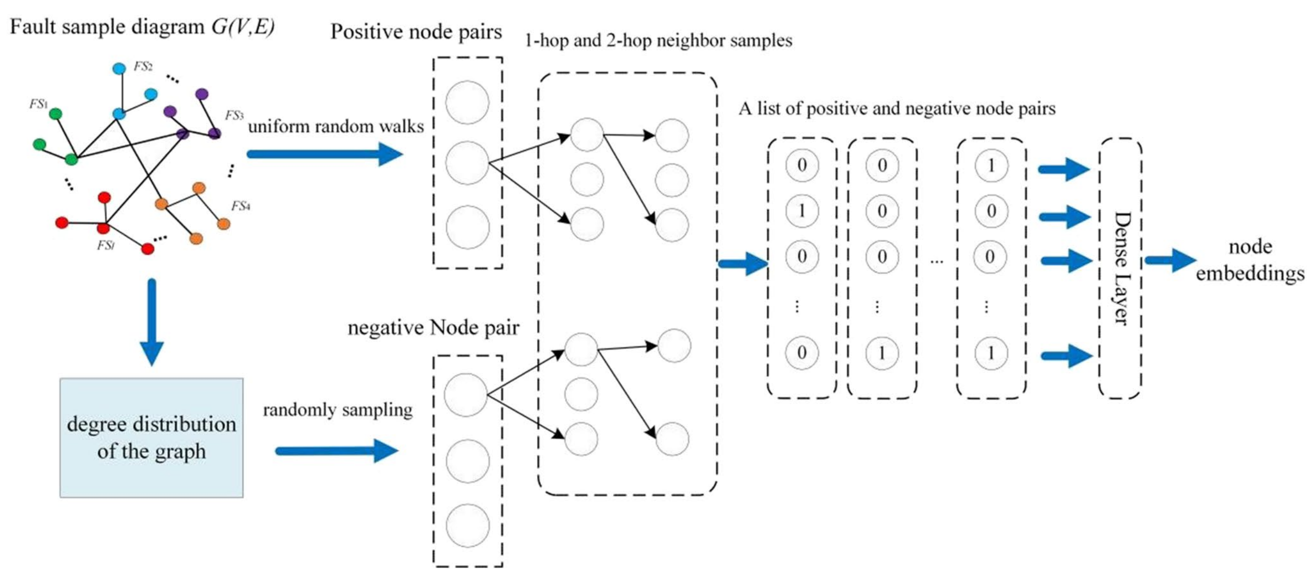


Fig. 3 Learing node embedding by GraphSAGE in an unsupervised way

Data processing First, vibration acceleration signals of various fault states are obtained. Then, signal samples $s = \{s_1, s_2, \dots, s_n\}$ can be acquired by uniform sampling to form the dataset S. Second, FFT is implemented on each signal sample to obtain their frequency feature $FFT(s) = \{FFT(s_1), FFT(s_2), \dots, FFT(s_n)\}$.

Graph construction The Euclidian distance between all signal samples is calculated based on the FFT feature. For each signal sample s_j, K nearest fault samples are selected as neighbors. Then, all signal samples are set as nodes with corresponding FFT features as node attributes, and each signal sample and its K nearest neighbors are connected to form edges. Then, a fault graph is constructed.

Unsupervised GraphSAGE model construction Based on the constructed fault graph, a two-layer GraphSAGE model with node sampling and feature aggregation is established to learn node embeddings. Then, a random walk is implemented on the fault graph, and connected node pairs (edges) are selected from the random walk and marked as positive. Then, unconnected node pairs are randomly selected from the graph with equal size and marked an negative. The embedding of positive and negative node pairs is calculated by concatenation based on the node embeddings. Then, they are fed into a dense layer to fulfill link prediction. With the aid of link labels, GraphSAGE can be trained in an unsupervised way. Finally, the feature of each signal sample can be extracted by training the proposed model.

Fault type identification A traditional classifier is implemented on the extracted features, which are split into training and testing sets to perform the fault identification task.

6 Experiments and data analysis

In this paper, we adopt two datasets to assess the proposed fault diagnosis method. First, a public dataset of rolling bearing fault diagnosis collected by the University of Paderborn, named the PU dataset [31], is adopted to evaluate the proposed method. Second, a fault diagnosis dataset collected by the constructed ZHS-2 motor rotor system, named the motor rotor dataset in this paper, is used as the assessment dataset. All the experiments are performed in the following environment: Windows 10, Intel Core i5-7300 @2.50 GHz. All the methods mentioned below are implemented in Python 3.7 and PyTorch 1.9.1.

6.1 Case 1: PU dataset

6.1.1 Data description

The PU dataset is a condition monitoring (CM) experimental rolling bearing dataset based on vibration and motor current signals. It adopts rolling bearings of Model 6023 manufactured by FAG, MTK and IBU to conduct malfunction tests. Table 2 demonstrates the PU dataset adopted in this paper. To be specific, three fault types, the normal operating conditions, outer ring faults and inner ring faults, are considered in this paper. Accordingly, they are specified in turn with label 0, label 1 and label 2. In this paper, an outer ring fault and an inner ring fault are real damage faults that are generated by accelerated life tests. The table lists the selected 16 groups of vibration signals, which are divided by fault type into three categories. The sampling

Table 1 The parameter settings of the classifiers

Classifier	General Introduction	Core Parameter Settings
Logistic Regression	It is a predictive analysis technique based on probability. It is used to predict the likelihood of classification dependent variables	Optimizer algorithm: solver = "lbfgs"(quasi-Newton method)
Decision Tree	It is the process of classifying instances based on a feature	The maximum depth of the decision tree: max_depth = None
Random Forest	It is an algorithm that integrates multiple trees through the idea of ensemble learning	The number of decision trees: n_estimators = 20
SVM	It is a kind of generalized linear classifier for classification of data according to supervised learning	Kernel function: kernel = 'rbf'
MLP	It is an artificial neural network with a forward structure, and a back propagation algorithm is generally used to train the multilayer perceptron	Optimizer algorithm: adam Iteration: max_iter = 1000
KNN	It is a nearest neighbor classification algorithm. All samples of known categories are taken as references to judge the category of unknown samples	Number of neighbors: n_neighbors = 5
GaussianNB	It is a generation method and is to directly find the joint distribution of the output and feature	-
AdaBoost	It is a boosting algorithm that combines multiple weak classifiers to create a strong ensemble classifier	Cycle index of the base classifier: n_estimators = 50, Learning rate: learning_rate = 0.001

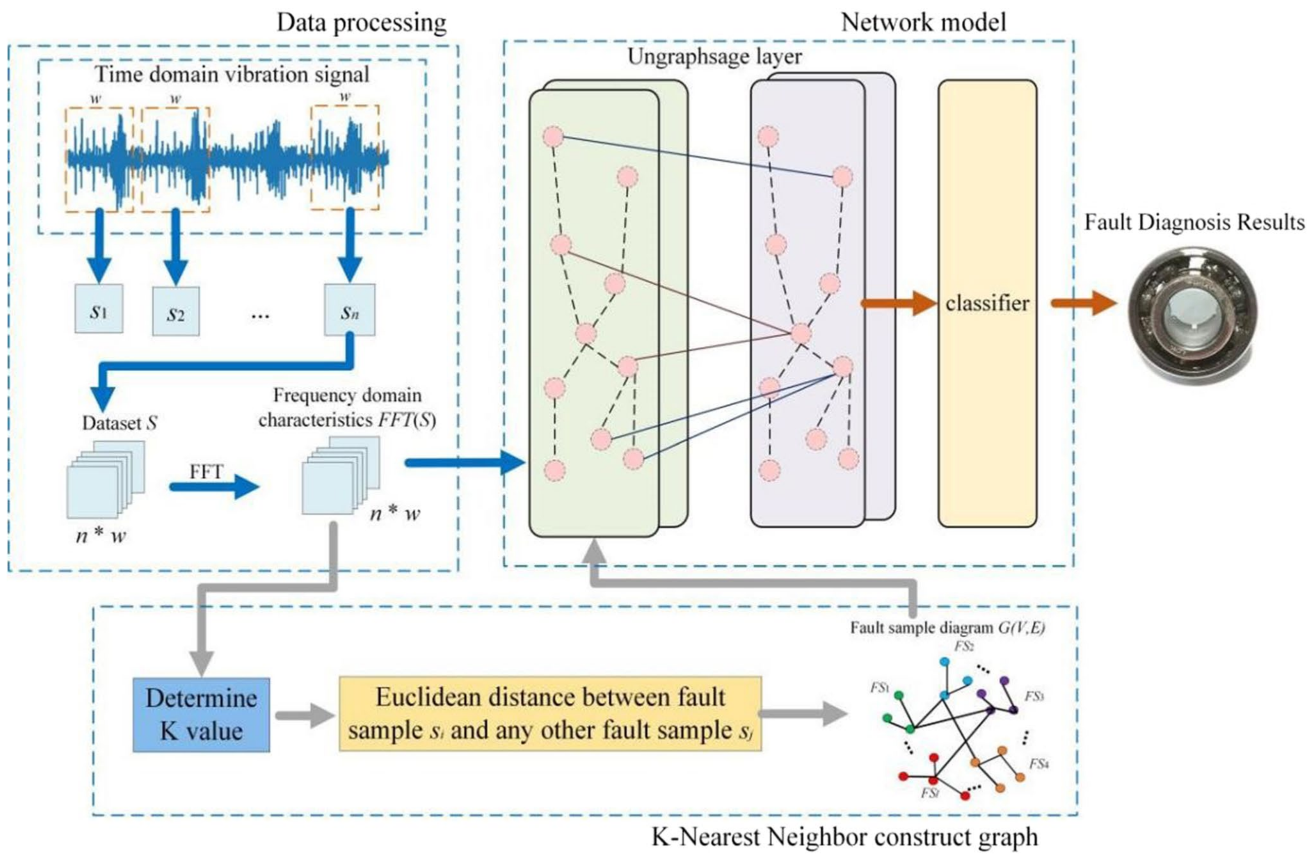


Fig. 4 Pipeline of fault diagnosis based on feature extraction via an unsupervised GraphSAGE model

frequency of the vibration signals is 64 kHz. All the signals are measured under the conditions of a driving system speed of 1500 RPM, a load torque of 0.7 nm and a radial force of 1000 N.

As shown in the table, the damage is mainly caused by fatigue and plastic deformation, which are presented as pitting and indentations, respectively, on the outer ring and inner ring of a rolling bearing. Furthermore, it can be seen that the same damage has different damage characteristics, damage combinations, damage arrangements and damage extents, and the details are shown in the table. It can be seen from the table that a total of 16 signals are selected in this paper. They are grouped into three categories: 5 signals under normal operating conditions, 5 signals under conditions of outer ring faults and 6 signals under conditions of inner ring faults.

In this paper, uniform sampling is adopted to collect the signal samples. Specifically, every 400 data points are considered a sample, and 500 samples are selected from each category. That is, a dataset with a total of $500 \times 3 = 1500$ fault samples is obtained. Then, the dataset is prepared by splitting it into a training set, validation set and test set in a certain proportion for further experiments.

6.1.2 Evaluation of feature extraction

In this section, we evaluate the feature extraction performance of unsupervised GraphSAGE in the fault diagnosis case on PU data. After implementing unsupervised GraphSAGE, signal samples are transformed into feature vectors. Then, t-SNE is adopted for visualization. Moreover, we compare our proposed method with two other feature extraction methods, FFT and an autoencoder. In detail, the frequency feature is obtained by implementing a FFT on each signal sample. The autoencoder we adopted for comparison is constructed with four linear layers in the encoder and decoder. Figure 5 demonstrates the t-SNE visualization of features extracted by the three methods: a) t-SNE is applied on the feature learned by the proposed method; b) t-SNE is directly applied on the frequency feature of the fault samples; c) t-SNE is applied on the feature learned by the autoencoder.

As shown in the figures, although the clusters of the three fault types contain a small number of mixed data points, the three clusters are independently separated. Thus, the feature learned by unsupervised GraphSAGE has a high degree of fault identification. In contrast, the three clusters in both Fig. 5(b) and Fig. 5(c) overlap

Table 2 Description of the PU dataset

Normal operating condition (Label 0)	Outer ring fault (Label 1)			Inner ring fault (Label 2)		
K001	KA04	Damage	fatigue: pitting	KI04	Damage	fatigue: pitting
		Characteristic of damage	single point		Characteristic of damage	single point
		Combination	single damage		Combination	multiple damage
		Arrangement	no repetition		Arrangement	no repetition
		Extent of damage	1		Extent of damage	1
K002	KA15	Damage	plastic deform: Indentations	KI14	Damage	fatigue: pitting
		Characteristic of damage	single point		Characteristic of damage	single point
		Combination	single damage		Combination	multiple damage
		Arrangement	no repetition		Arrangement	no repetition
		Extent of damage	1		Extent of damage	1
K003	KA16	Damage	fatigue: pitting	KI16	Damage	fatigue: pitting
		Characteristic of damage	repetitive damage		Characteristic of damage	single point
		Combination	no repetition		Combination	single damage
		Arrangement	random		Arrangement	no repetition
		Extent of damage	2		Extent of damage	3
K004	KA22	Damage	fatigue: pitting	KI17	Damage	fatigue: pitting
		Characteristic of damage	single point		Characteristic of damage	single point
		Combination	single damage		Combination	repetitive damage
		Arrangement	no repetition		Arrangement	random
		Extent of damage	1		Extent of damage	1
K005	KA30	Damage	plastic deform: Indentations	KI18	Damage	fatigue: pitting
		Characteristic of damage	distributed		Characteristic of damage	single point
		Combination	repetitive damage		Combination	single damage
		Arrangement	random		Arrangement	no repetition
		Extent of damage	1		Extent of damage	2
—-	—-			KI21	Damage	fatigue: pitting
					Characteristic of damage	single point
					Combination	single damage
					Arrangement	no repetition
					Extent of damage	1

Damage: 1. fatigue: pitting; 2. plastic deform: Indentations

Characteristic of damage: 1. single point; 2. distributed

Combination: 1. single damage: A single component of a rolling bearing is affected by a single damaged area; 2. repetitive damage: A single component of a rolling bearing is affected by a single damages area and the same damage symptoms are repeated in several places of the same bearing part; 3. multiple damage: Different breakage symptoms occur on different parts of the same bearing

Arrangement: 1. no repetition: The breach occurred only once; 2. Random: Damage symptoms are distributed in a random fashion across the components

Extent of damage: 1. Limit for a bearing is less than or equal to 2 mm; 2. Limit for a bearing is more than 2 mm;3. Limit for a bearing is more than 4.5 mm

to some extent. It can be concluded that the features extracted by the two comparative methods present weaker fault identification abilities.

6.1.3 Fault diagnosis accuracy

As illustrated in Sect. 3, traditional classifiers are implemented on the extracted feature for further fault diagnosis. As shown in Table 3, unsupervised GraphSAGE is combined with 8 classifiers: logistic regression, decision tree, random forest, SVM, MLP, KNN, Gaussian NB and AdaBoost classifiers.

All eight methods are tested under the same conditions. Specifically, signal samples of the PU dataset are split into training and testing sets, and the rate of the training set decreases from 0.9 to 0.1; accordingly, the rate of the

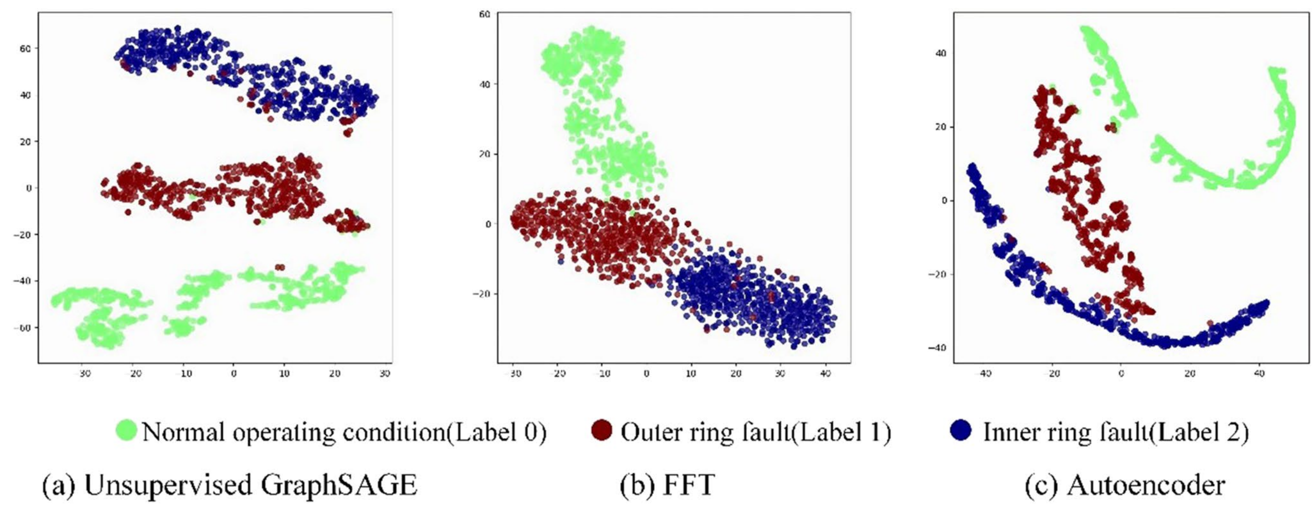


Fig. 5 Feature visualization of the PU dataset. **a** Unsupervised GraphSAGE, **b** FFT, **c** Autoencoder

testing set increases from 0.1 to 0.9. The fault diagnosis accuracies of eight methods on the PU dataset are given in the table. As shown in the table, except for the AdaBoost-based method, the accuracy of all the other methods is above 90% for all combinations of training and testing sets. Moreover, for the remaining 7 methods, the accuracy of the decision tree classifier is approximately 2 ~ 3% lower than that of the other methods. When the rate of the training set is decreased, the accuracies of the random forest and GaussianNB classifiers are slightly reduced. Thus, it can be concluded that logistic regression, SVM, MLP and KNN have equally good performance on the fault diagnosis of PU data.

Moreover, to further verify the performance of the proposed fault diagnosis method, the proposed method is compared with some related traditional fault diagnosis methods: a

Table 3 Fault diagnosis accuracy on the PU dataset based on unsupervised GraphSAGE combined with different classifiers

Train_rate	Test_rate	Unsupervised GraphSAGE + Logistic Regression	Unsupervised GraphSAGE + Decision Tree	Unsupervised GraphSAGE + Random Forest	Unsupervised GraphSAGE + SVM
0.9	0.1	0.9791	0.9509	0.9738	0.9787
0.8	0.2	0.9758	0.9493	0.9756	0.9763
0.7	0.3	0.9759	0.9513	0.9742	0.9783
0.6	0.4	0.9741	0.9463	0.9737	0.9804
0.5	0.5	0.9756	0.9427	0.9729	0.9799
0.4	0.6	0.9748	0.9411	0.9709	0.9768
0.3	0.7	0.9750	0.9400	0.9703	0.9730
0.2	0.8	0.9738	0.9297	0.9681	0.9729
0.1	0.9	0.9737	0.9182	0.9679	0.9746
Train_rate	Test_rate	Unsupervised GraphSAGE + MLP	Unsupervised GraphSAGE + KNN	Unsupervised GraphSAGE + GaussianNB	Unsupervised GraphSAGE + AdaBoost
0.9	0.1	0.9711	0.9778	0.9831	0.7684
0.8	0.2	0.9736	0.9733	0.9743	0.7727
0.7	0.3	0.9719	0.9754	0.9747	0.7856
0.6	0.4	0.9757	0.9769	0.9763	0.7248
0.5	0.5	0.9749	0.9774	0.9780	0.7587
0.4	0.6	0.9745	0.9748	0.9753	0.7837
0.3	0.7	0.9716	0.9706	0.9678	0.7713
0.2	0.8	0.9712	0.9708	0.9690	0.7825
0.1	0.9	0.9724	0.9714	0.9691	0.8204

Table 4 Parameter settings of the compared methods

Methods	Hyperparameters
1D_CNN	learning_rate = 0.001, epochs = 100, n_layer = 3, kernel_size = 2
GCN	neighbors = 5, learning_rate = 0.001, epochs = 100, hidden_layer = 200, n_layer = 2
LSTM	learning_rate = 0.001, epochs = 100, n_layer = 4, hidden_dim = 100, batch_size = 50
AE + MLP	learning_rate = 0.001, epochs = 1000, n_layer = 4

convolutional neural network (1D_CNN), a graph convolutional network (GCN) [24], long short-term memory (LSTM) and an autoencoder combined with an MLP classifier (AE + MLP). Table 4 demonstrates the parameter settings of the four comparison methods. Specifically, 1D_CNN is constructed with three combination layers, and each layer consists of a convolutional layer, a max pooling layer and a ReLU layer. The GCN has the same structure as as that in [28]. LSTM consists of four layers, and the dimension of the hidden layer is set as 100. The autoencoder is the same as that in the last experiment with four linear layers in the encoder and decoder. 1D_CNN, LSTM and the autoencoder are directly implemented on the signal samples. The GCN is implemented on the fault graph, which is constructed in the same way as that used by the proposed method, and the number of neighbors is set as 5.

Table 5 demonstrates the comparison results of the models in terms of accuracy. Similarly, signal samples are split into training and test sets, and the rates of both sets vary in the same way as in the last experiment. As shown in the table, the proposed method (unsupervised GraphSAGE + MLP) outperforms the other four methods. In particular, when few labeled data are available, the accuracy of the other four methods decreases more or less, but the accuracy of the proposed method remains at a steady level.

6.1.4 Parameter evaluation

Graph construction is a key process in the proposed method. In this paper, KNN is adopted to construct the graph, and *K*, which stands for the number of neighbors, is an important parameter. In this section, we evaluate the proposed method with different values of K when the rate of the training set is 0.9 and the test set is 0.1. As seen from the chart, *K* increases from 1 to 10. The triangle curve shows the results on the PU data. It can be concluded that the accuracy is relatively stable when the parameter is between 2 and 10, but when $K = 5$, the proposed method achieves the best result (Fig. 6).

6.2 Case 2: Motor rotor dataset

6.2.1 Data description

In this paper, a ZHS-2 motor rotor system is adopted to collect the motor rotor dataset. As shown in Fig. 7, a ZHS-2 motor rotor system with a flexible rotor is adopted as a test bench. In the system, 8 sensors are installed at different positions on the test bench to collect vibration acceleration signals. The HG-8902 data collection box is responsible for transmitting the collected signals. In this paper, 7 typical faults are considered: normal operation (F_1), loose fan rotor base (F_2), rotor imbalance I (F_3), rotor imbalance III (F_4), rotor imbalance V (F_5), rotor imbalance VII (F_6) and broken blade (F_7). Four kinds of rotor imbalance faults can be simulated by installing different numbers of screws on the rotor. For example, installing three screws simulates the rotor unbalance III fault.

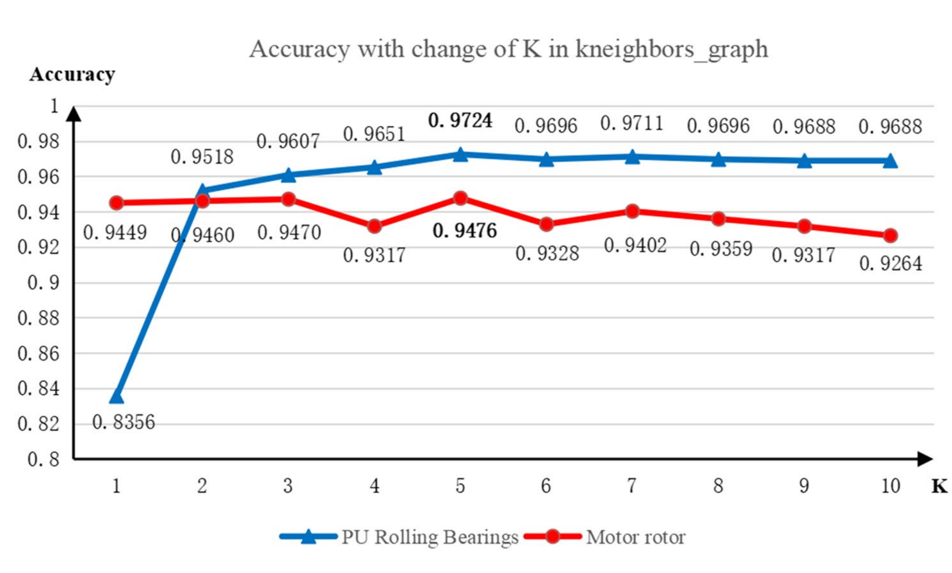
When running the test bench, the rotor speed is 1500 r/m, the sampling frequency is 1280 Hz, the fundamental frequency *1X* is 25 Hz, and the n-octave nX ($n = 1,2,3$) is ($n \times 25$) Hz. The faults cause abnormal vibration of the motor rotor, which is manifested in large or small changes

Table 5 Comparison of fault diagnosis methods on the PU dataset

Train_rate	Test_rate	Unsupervised GraphSAGE + MLP	1D_CNN	LSTM	AE + MLP	GCN
0.9	0.1	0.9711	**0.9733**	**0.9733**	0.9467	0.9667
0.8	0.2	**0.9736**	0.9667	0.9667	0.9567	0.9600
0.7	0.3	**0.9719**	0.9644	0.9633	0.9600	0.9689
0.6	0.4	**0.9757**	0.9467	0.9533	0.9633	0.9533
0.5	0.5	**0.9749**	0.9467	0.9633	0.9653	0.9453
0.4	0.6	**0.9745**	0.9567	0.9640	0.9656	0.9522
0.3	0.7	**0.9716**	0.9467	0.9644	0.9590	0.9562
0.2	0.8	**0.9712**	0.9408	0.9495	0.9625	0.9542
0.1	0.9	**0.9724**	0.9311	0.9442	0.9548	0.9452

The bold entries means the best diagnosis accuracy

Fig. 6 Evaluation of K in the KNN graph



in the vibration amplitude of the signals. Therefore, the FFT is adopted to extract the frequency feature of the signals. To conclude, 300 fault samples are sampled for each fault, and a total of $300 \times 7 = 2100$ fault samples are obtained. For each fault sample, the amplitude of the $1X \sim 3X$ octave is extracted as a feature, and a total of $3 \times 8 = 24$ features are acquired. Therefore, fault data with 2100×24 are obtained, and then a fault graph with 2100 nodes can be constructed with fault samples as nodes and the FFT features as node attributes. Similar to the PU dataset, the proposed method is evaluated on the motor rotor dataset from three aspects.

6.2.2 Evaluation of feature extraction

Similarly, we evaluate the feature extraction ability of the proposed method by applying t-SNE to visualize the embedding of the fault samples in this section. Figure 8 demonstrates the t-SNE visualization of features extracted by the proposed method and two other comparison methods: a) t-SNE is applied on the feature learned by the proposed method; b) t-SNE is directly applied on the FFT feature of the fault samples; and c) t-SNE is applied on the feature learned by the autoencoder. For all three figures, the seven fault types are represented by different colors. Obviously, the result of the FFT is far from acceptable, as there is a certain mixture area in the figure. By observing the figures of the proposed method and the autoencoder, although there are mixed points on both figures, the five clusters in Fig. 8(a) are slightly more divergent than those in Fig. 8(c).

6.2.3 Fault diagnosis accuracy

In this section, experiments on the motor rotor dataset are set in the same way as those on the PU dataset. Compared

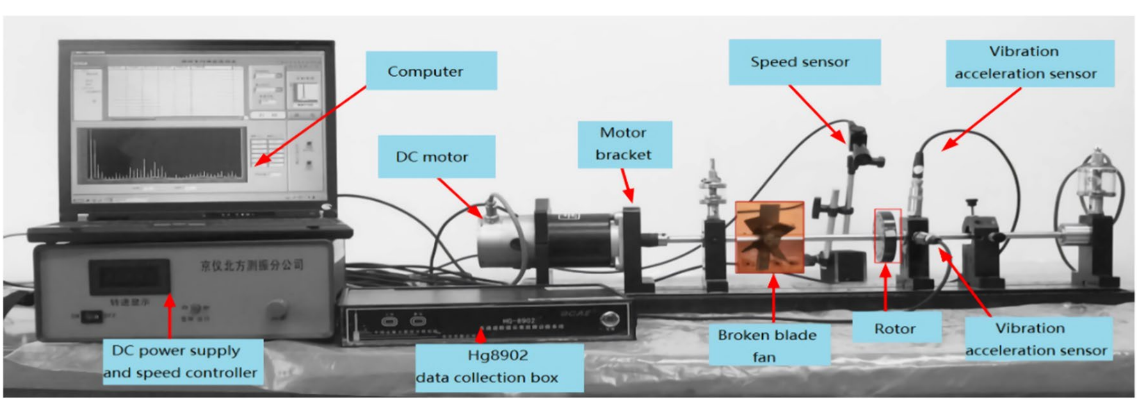


Fig. 7 ZHS-2 motor rotor system

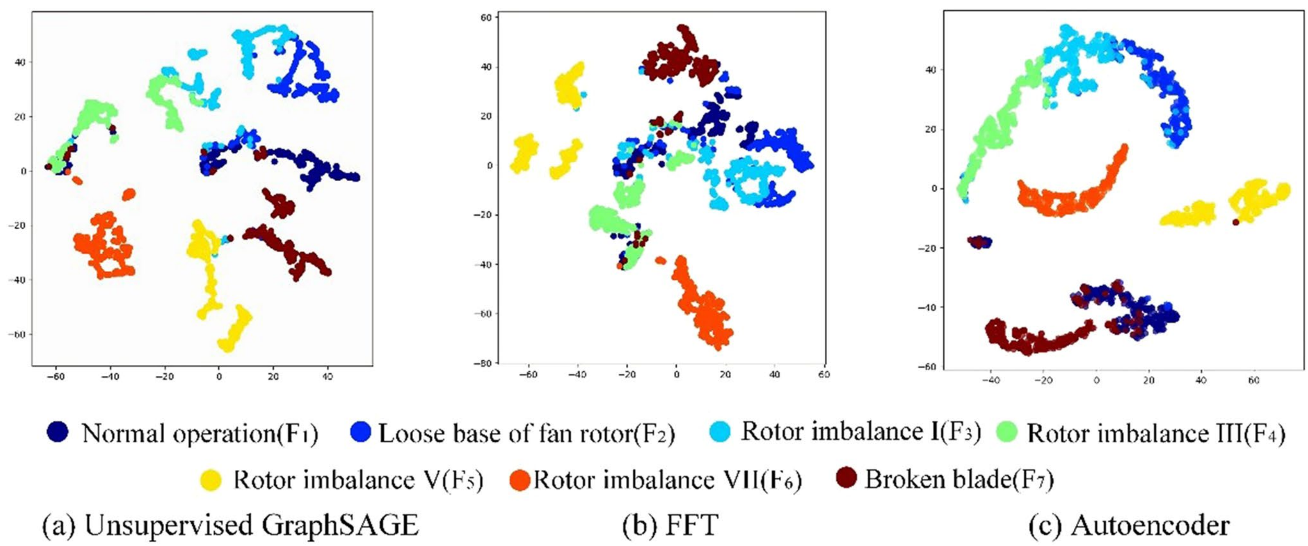


Fig. 8 Feature visualization of the motor rotor dataset. **a** Unsupervised GraphSAGE, **b** FFT, **c** Autoencoder

with the PU dataset, the motor rotor dataset has more fault types. The increase in the class number is a kind of challenge for some classifiers. Table 6 lists the fault diagnosis accuracy of the motor rotor dataset with unsupervised GraphSAGE combined with eight different classifiers. The results in the table show that AdaBoost does not work in the case of fault diagnosis on the motor rotor dataset. Moreover, the accuracy of the decision tree-based method is lower than that of the other six methods. In particular, when there are fewer training data, its accuracy decreases to approximately 80%. Furthermore, the logistic regression, random forest, SVM, and KNN-based methods have equal fault diagnosis performance on the motor rotor dataset. Finally, it can be concluded that the MLP-based fault diagnosis method has

Table 6 Fault diagnosis accuracy of the motor rotor dataset with unsupervised GraphSAGE combined with different classifiers

Train_rate	Test_rate	Unsupervised GraphSAGE + Logistic Regression	Unsupervised GraphSAGE + Decision Tree	Unsupervised GraphSAGE + Random Forest	Unsupervised GraphSAGE + SVM
0.9	0.1	0.9576	0.9258	0.9668	0.9682
0.8	0.2	0.9603	0.9265	0.9631	0.9704
0.7	0.3	0.9603	0.9215	0.9610	0.9683
0.6	0.4	0.9563	0.9169	0.9591	0.9687
0.5	0.5	0.9527	0.9112	0.9560	0.9602
0.4	0.6	0.9500	0.8976	0.9540	0.9445
0.3	0.7	0.9427	0.8832	0.9504	0.9445
0.2	0.8	0.9358	0.8669	0.9420	0.9352
0.1	0.9	0.9163	0.8146	0.9175	0.9295
Train_rate	Test_rate	Unsupervised GraphSAGE + MLP	Unsupervised GraphSAGE + KNN	Unsupervised GraphSAGE + GaussianNB	Unsupervised GraphSAGE + AdaBoost
0.9	0.1	0.9830	0.9636	0.9188	0.4661
0.8	0.2	0.9869	0.9669	0.9212	0.4531
0.7	0.3	0.9859	0.9658	0.9233	0.4273
0.6	0.4	0.9811	0.9619	0.9175	0.4374
0.5	0.5	0.9790	0.9573	0.9187	0.4198
0.4	0.6	0.9711	0.9476	0.9107	0.4663
0.3	0.7	0.9684	0.9462	0.9126	0.4518
0.2	0.8	0.9546	0.9284	0.9081	0.4663
0.1	0.9	0.9401	0.9019	0.8998	0.4762

Table 7 Comparison of fault diagnosis methods on the motor rotor dataset

Train_rate	Test_rate	Unsupervised GraphSAGE +MLP	1D_CNN	LSTM	AE+MLP	GCN
0.9	0.1	**0.9830**	0.9667	0.9571	0.9476	0.8714
0.8	0.2	**0.9869**	0.9595	0.9476	0.9262	0.8667
0.7	0.3	**0.9859**	0.9524	0.9404	0.9270	0.8571
0.6	0.4	**0.9811**	0.9250	0.9222	0.9333	0.8464
0.5	0.5	**0.9790**	0.9429	0.9047	0.9314	0.8352
0.4	0.6	**0.9711**	0.9484	0.8923	0.9278	0.8389
0.3	0.7	**0.9684**	0.9578	0.8651	0.9327	0.8395
0.2	0.8	**0.9546**	0.9321	0.8184	0.9393	0.8500
0.1	0.9	**0.9401**	0.9254	0.6839	0.9190	0.8249

The bold entries means the best diagnosis accuracy

distinct advantages, achieving the best accuracy at all training set rates.

Table 7 displays the accuracy of the comparison fault diagnosis methods on the motor rotor dataset. From the table, we can see that the proposed method (unsupervised GraphSAGE + MLP) outperforms the other four deep learning-based methods for all training and testing set rates. In particular, when few labeled data are available, the proposed method can still reach a precise diagnosis result.

6.2.4 Parameter evaluation

The parameter K of KNN for graph construction is also evaluated for the motor rotor data. As shown in Fig. 6, the dot curve shows the results on the motor rotor data. In the curve, K increases from 1 to 10, and the accuracy is relatively stable in the whole parameter range. However, when $K = 5$, the accuracy is slightly higher than at the other values.

7 Conclusions

In this paper, a fault diagnosis method based on feature extraction via an unsupervised graph neural network is proposed. First, signal samples of various fault types are obtained by applying uniform sampling to the collected vibration signals. Then, the FFT is implemented on each sample to extract frequency features. Based on all signal samples and their frequency features, KNN is adopted to construct a fault graph that conveys both feature and relation information of signal samples. Then, GraphSAGE is implemented on the graph in an unsupervised way to achieve feature extraction; thus, features of each signal sample can be obtained. Finally, some traditional classifiers are used to identify fault states by analyzing the learned features. Compared with some traditional deep learning-based methods, such as CNN, LSTM, GCN and autoencoder methods, the proposed method can acquire better identification accuracy. Typically, when only a few labeled samples are available, the proposed method can still achieve an accurate fault diagnosis result. Moreover, graph construction plays a crucial role in the proposed model. The K-nearest neighbor algorithm is implemented on the FFT feature to construct a graph in this paper, which requires a large amount of computation and may suffer from high-dimensional problems. In future work, a more efficient graph construction method can be researched.

Acknowledgements We acknowledge financial support from the National key R&D project (2022YFE0210700), the Zhejiang Province Public Welfare Technology Application Research Project (LGF21F020013), Zhejiang Province Outstanding Youth Fund (LR21F030001), Zhejiang Province Key R&D projects (2021C03142, 2021C03015), National Natural Science Foundation of China General Program (52171352), and National Natural Science Foundation of China (62103121).

Author Contribution Jing Feng(Conceptualization/Writing-Original Draft), Shouyang Bao (Investigation), Xiaobin Xu(Methodology), Zhenjie Zhang(Data Curatuion), Pingzhi Hou(Supervision), Felix Steyskal(Writing-Original Draft), Schahram Dustdar(Writing-Review&Editing).

Data availability The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Conflict of Interest We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abid A, Khan MT, Ipbal J (2021) A review on fault detection and diagnosis techniques: basics and beyond. Artif Intell Rev 54:3639–3664

2. Gangsar P, Tiwari R (2020) Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. Mech Syst Signal Process 144:106908
3. Abid A, Khan MT, Khan MS (2020) Multidomain features-based GA optimized artificial immune system for bearing fault detection. IEEE Trans Syst Man Cybern Syst 50(1):348–359
4. Li DM, Cai ZM, Qin B et al (2020) Signal frequency domain analysis and sensor fault diagnosis based on artificial intelligence. Comput Commun 160:71–80
5. Jiang ZH, Zhang K, Xiang L et al (2023) A time-frequency spectral amplitude modulation method and its applications in rolling bearing fault diagnosis. Mech Syst Signal Process 185:109832
6. Minhas AS, Kankar PK, Kumar N et al (2021) Bearing fault detection and recognition methodology based on weighted multiscale entropy approach. Mech Syst Signal Process 147:107073
7. Zhao B, Zhang XM, Li H, Yang ZB (2020) Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions[J]. Knowl-Based Syst 199:105871
8. Zhang YH, Zhou TT, Huang XF, Cao LC, Zhou Q (2021) Fault diagnosis of rotating machinery based on recurrent neural networks. Measurement 171:108774
9. Su XY, Cao CQ, Zeng XD et al (2021) Application of DBN and GWO-SVMin analog circuit fault diagnosis. Sci Rep 11(1):7969–7969
10. Yang Z, Xu BB, Luo W et al (2022) Autoencoder-based representation learning and its application in intelligent fault diagnosis: A review. Measurement 189:110460
11. Kipf TN, Welling M (2017) Semi-Supervised Classification with Graph Convolutional Networks. In International Conference on Learning Representations, April 24–26, Toulon, France
12. Grover A and Leskovec J, Node2vec: Scalable Feature Learning for Networks, In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, August 13–17, San Francisco California USA
13. Hamilton WL, Ying R, Leskovec J. Inductive Representation Learning on Large Graphs, In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS) ,2017, December 4 - 9, Long Beach, CA, USA
14. Chen W, Li JN, Wang Q (2021) Fault feature extraction and diagnosis of rolling bearings based on wavelet thresholding denoising with CEEMDAN energy entropy and PSO-LSSVM. Measurement 172:108901
15. Boztas G, Tuncer T (2022) A fault classification method using dynamic centered one-dimensional local angular binary pattern for a PMSM and drive system. Neural Comput Appl 34:1981–1992
16. Zhang JQ, Zhang Q, Qin XR et al (2022) A two-stage fault diagnosis methodology for rotating machinery combining optimized support vector data description and optimized support vector machine. Measurement 200:111651
17. He C, Wu T, Gu R, Jin Z, Ma R, Qu H (2021) Rolling bearing fault diagnosis based on composite multis cale permutation entropy and reverse cognitive fruit fly optimization algorithm – Extreme learning machine. Measurement 173:108636
18. Si L, Wang Z, Tan C, Liu X (2019) A feature extraction method based on composite multi-scale perm utation entropy and Laplacian score for shearer cutting state recognition. Measurement 145:84–93
19. Yang C, Jia M (2021) Hierarchical multiscale permutation entropy-based feature extraction and fuzzy support tensor machine with pinball loss for bearing fault identification. Mech Syst Signal Process 149:107182
20. Leite G, Araújo A, Rosas P et al (2019) Entropy measures for early detection of bearing faults. Physica A 514:458–472
21. Liu ZL, Fang LL, Jinag D, Qu QH (2022) A machine-learning-based fault diagnosis method with adaptive secondary sampling for multiphase drive Systems. IEEE Trans Power Electron 37(8):8767–8772
22. Li X, Li X, Ma H (2020) Deep representation clustering-based fault diagnosis method with unsupervised data applied to rotating machinery. Mech Syst Signal Process 143:106825
23. Xiao DY, Qin CJ, Yu HJ et al (2021) Unsupervised machine fault diagnosis for noisy domain adaptation using marginal denoising autoencoder based on acoustic signal. Measurement 176:109186
24. Xia M, Li T, Liu LZ et al (2017) Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder. IET Sci Meas Technol 11(6):687–695
25. Luo SY, Huang XF, Wang YZ et al (2022) Transfer learning based on improved stacked autoencoer for bearing fault diagnosis. Knowl-Based Syst 256:109846
26. Veličković P, Cucurull G, Casanova A. Graph Attention Networks. International Conference on Learning Representations (ICLR), 2018, April 30- May 3, Vancouver Canada
27. Li T, Zhou Z, Li S, Sun C, Yan R, Chen X (2022) The emerging graph neural networks for intelligent fault diagnostics and prognostics: a guideline and a benchmark study. Mech Syst Signal Process 168:108653
28. Gao YY, Chen M, Yu DJ (2021) Semi-supervised graph convolutional network and its application in intelligent fault diagnosis of rotating machinery. Measurement 186:110084
29. Kavianpour M, Ramezani A, Beheshti MTH (2022) A class alignment method based on graph convolution neural network forbearing fault diagnosis in presence of missing data and changing working conditions. Measurement 199:111536
30. Abeywickrama T, Cheema MA, Taniar D (2016) k-Nearest Neighbors on Road Networks: A Journey in Experimentation and In-Memory Implementation. Proc VLDB Endowment 9(6):492–503
31. Zhu ZQ, Lei YB, Qi GQ et al (2023) A review of the application of deep learning in intelligent fault diagnosis of rotating machinery. Measurement 206:112346

Jing Feng was born in Xi'an, Shaanxi, China in 1984. She received the B.S. in electronic engineering and M.S. degrees in pattern recognition and intelligent system from the Xidian University in 2007 and 2010 respectively, and the Ph.D. degree in computer science from University of Munich in 2014. Since 2019, she has been a lecturer with School of Automation (School of Artificial Intelligence) and the Belt and Road information technology research institute, Hangzhou Dianzi University, Hangzhou, China. Her research interests include fault diagnosis, graph neural network and graph mining.

Shouyang Bao was born in Jilin, China in 1999. He received the B.E. degree in electronic information engineering from the Changchun University in 2021. Since 2021, he has been a postgraduate student with School of Automation (School of Artificial Intelligence), Hangzhou Dianzi University, Hangzhou, China. His main research field is fault diagnosis.

Xiaobin Xu received the Ph.D. degree in power electronics and power transmission from Shanghai Maritime University, China, in 2009. He is a professor of department of automation and the Belt and Road information technology research institute in Hangzhou Dianzi University.

His research interests include evidence theory, fuzzy set theory and applications in the processing of uncertain information, the reliability analysis, safety evaluation and condition monitoring of complex industrial systems.

Zhenjie Zhang was born in Haining, Zhejiang, China in 1989. He received the B.S. and M.S. degrees in mechanical engineering from the Hangzhou Dianzi University in 2012 and 2015 respectively, and the Ph.D. degree in mechanical engineering from Zhejiang University of Technology in 2019. Since 2019, he has been a lecturer with School of Automation (School of Artificial Intelligence) and the Belt and Road information technology research institute, Hangzhou Dianzi University, Hangzhou, China. His research interests include uncertain information fusion, fault diagnosis and complex networks.

Pingzhi Hou was born in Henan, China, in November 1968. He received B.E. degree in production process automation from Wuhan Institute of Technology, Wuhan, China, in 1990. He is a professor of department of automation in Hangzhou Dianzi University. His research interests include fault detection and industry automation.

Felix Steyskal received his Master and PhD degree in Cultural technology & water management on the University of life science, Vienna, Austria from 1981 to 1991. He was the Head of the Institute of Agrobiotechnology Tulln, Austria, Leading a team of 120 technicians and scientists in the field of agrobiotechnology from 1995 to 2004, and the Head of Department Environmental Resources & Technologies, Center of Energy of the Austrian Institute of Technology. He is the CTO of M-U-T GmbH.