Front. Mech. Eng. 2023, 18(2): 20
https://doi.org/10.1007/s11465-022-0736-9

RESEARCH ARTICLE

A multi-sensor relation model for recognizing and localizing faults of machines based on network analysis

Shuhui WANG[a], Yaguo LEI (✉)[a], Na LU[b], Xiang LI[a], Bin YANG[a]

[a] *Key Laboratory of Education Ministry for Modern Design and Rotor-Bearing System, Xi'an Jiaotong University, Xi'an 710049, China*
[b] *Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, China*

✉ *Corresponding author. E-mail: yaguolei@mail.xjtu.edu.cn (Yaguo LEI)*

ABSTRACT Recently, advanced sensing techniques ensure a large number of multivariate sensing data for intelligent fault diagnosis of machines. Given the advantage of obtaining accurate diagnosis results, multi-sensor fusion has long been studied in the fault diagnosis field. However, existing studies suffer from two weaknesses. First, the relations of multiple sensors are either neglected or calculated only to improve the diagnostic accuracy of fault types. Second, the localization for multi-source faults is seldom investigated, although locating the anomaly variable over multivariate sensing data for certain types of faults is desirable. This article attempts to overcome the above weaknesses by proposing a global method to recognize fault types and localize fault sources with the help of multi-sensor relations (MSRs). First, an MSR model is developed to learn MSRs automatically and further obtain fault recognition results. Second, centrality measures are employed to analyze the MSR graphs learned by the MSR model, and fault sources are therefore determined. The proposed method is demonstrated by experiments on an induction motor and a centrifugal pump. Results show the proposed method's validity in diagnosing fault types and sources.

KEYWORDS fault recognition, fault localization, multi-sensor relations, network analysis, graph neural network

1 Introduction

With the rapid development of Internet of Things techniques, distributed monitoring sensors are increasingly used in mechanical systems' surveillance, resulting in a large volume of multivariate sensing data [1,2]. These multivariate sensing data can provide not only signals but also the relations among sensors. Meanwhile, components of modern industrial equipment are enabled to communicate with one another since they are embedded with various sensors [3]. The emerging interconnection of the components brings a fresh perspective to making decisions for machine fault diagnosis over multivariate sensing data, namely, utilizing multi-sensor relations (MSRs). Although previous works [4,5] have reported that MSRs improve diagnosis accuracy, fault diagnosis should benefit more than just accuracy-enhancing from MSRs. Exisiting studies have exerted much effort to enhance the diagnosis accuracy of recognizing fault types [6–9] while localizing fault sources is seldom investigated. Such localization is crucial in conducting effective and efficient maintenance plans [10], especially for machines with multiple sources [11,12]. For an induction motor, a specific fault type may be caused by any motor phase (i.e., a fault type corresponds to multiple sources). Identifying the potential source of that fault type out of multiple sources is of great importance. In these scenarios, locating fault sources are as important as recognizing fault types. Therefore, it is urgent to develop a global method that can achieve fault recognition and fault localization simultaneously. In this article, fault recognition refers to recognizing different fault types, and fault localization describes identifying the potential source of certain faults.

Currently, the mainstream in multi-sensor-based fault diagnosis is to take advantage of the complementary information provided by different sensors while removing their redundancy and contradiction [13]. Previous research has shown that multi-sensor fusion can be broken down into three levels: data-level fusion, feature-level fusion, and decision-level fusion. The strength of data-level fusion is to use all the sensor information directly by concatenating the raw data [14]. Feature fusion level fuses high-level features extracted from the

Received April 17, 2022; accepted August 22, 2022

raw data through various techniques [15]. Decision-level fusion aims to overcome the conflict between the sensors, like evidence reasoning [16] or voting strategy [17]. As a result, multi-sensor fusion is preferred due to the enrichment of different sensor information to achieve high diagnosis accuracy for fault recognition and fault localization [18,19]. However, previous research has neglected utilizing MSRs. Recently, graph neural networks (GNNs) have received increasing attention in exploiting MSRs for fault diagnosis. Zhou et al. [4] constructed graph data for multi-sensor vibration signals and then fed the graph data to a graph convolutional network (GCN) for fault recognition in a gearbox. Li et al. [5] generated spatial-temporal graphs from the multivariate temporal data and then used the graphs for graph learning to improve the remaining useful life prediction. However, they only employ MSRs for fault recognition or fault prognostic rather than for fault localization. To localize faults, similarity metric based methods have been widely studied . With the Euclidean distance, Pearson correlation coefficient, and cosine similarity calculated from three-phase currents, Wu and Zhao [20] localized the broken leg of the faulty transistor to realize an open-circuit fault diagnosis. Irhoumah et al. [21] calculated the Pearson correlation coefficient from multivariate sensing data for short-circuit fault diagnosis of asynchronous motors. Despite the great success in localizing faults using similarity metrics, they generally diagnose a single fault type (e.g., the open-circuit fault). Meanwhile, similarity metrics are sensitive to noise when calculating MSRs [22], whereas the collected multivariate sensing data are usually noisy.

Considering the limitations of existing approaches, this article proposes using GNNs to achieve fault recognition of multiple fault types as well as fault localization with MSRs. However, the graph learning process of existing GNNs often assumes explicit graph structures as inputs. In the fields with explicit graph structures, GNNs can be directly used. Otherwise, an explicit graph structure should be calculated in advance. For multi-sensor-based fault diagnosis, the application of GNNs has two issues to be addressed: 1) No explicit graph structure is available for MSRs, and 2) the calculation for precise MSRs is not always easy because of the harsh working environment of machines. To address the aforementioned issues, we propose an MSR model incorporating self-learning graphs into the graph-learning process. The model is expected to provide self-learned graphs for the MSRs, as opposed to those calculated by similarity metrics. The main contributions are highlighted as follows.

First, a two-stage global fault diagnosis (GFD) method is proposed. The first stage is to recognize the fault types of machines. The second stage is to localize the possible source of certain fault types. In the second stage, centrality measures are calculated for network analysis to determine the critical node of MSRs.

Second, an MSR model is established. The MSR model integrates the calculation for MSRs and learning from MSRs in a whole graph learning process. Thus, the situation with no explicit graph structure for MSRs can be solved.

The rest of the article is organized as follows. Section 2 gives an overview of the proposed method as well as the details of the MSR model. Section 3 provides two multi-source fault cases to verify the effectiveness proposed method. Section 4 summarizes the conclusions.

2 Proposed GFD method

Graphs are mathematical abstracts to model relations between objects. In this article, sensors are nodes, and their relations are edges. Modeling MSRs is equivalent to figuring out the explicit graph structure between sensor nodes. This work draws on ideas from graph learning to model and learn from MSRs.

In the rest of this section, an overview of the GFD method is given in Section 2.1, and the details of the MSR model will be elaborated on in Section 2.2.

2.1 Overview

The GFD method aims to achieve fault recognition and localization. The goal of fault recognition is to predict the correct label from the online monitoring data. The goal of fault localization is to localize the possible source of certain fault types. The schematic diagram of the GFD method is depicted in Fig. 1. An MSR model is established first by offline training. As shown in Fig. 1, the inputs to train the MSR model are multi-sensor signals rather than pre-defined graphs. Once the model is trained, both the fault types and their corresponding MSR graphs can be obtained. The current state of machines is justified in the stage of fault recognition. The trained model can output its fault type given the online monitoring data. After that, the source is identified in the stage of fault localization. In this stage, a two-step process is conducted. The first step is to get its explicit graph structure using the trained model. The second step is to conduct graph analytics on the self-learned graph using tools from network analysis [23,24].

To understand the proposed method better, some basic concepts of graphs are given below.

Definition I *Graph*. A graph is formulated as $G = (V, E, \boldsymbol{A})$, where V is the node (i.e., vertex) set, $E = \left\{ e_{ij} \middle| (v_i, v_j) \in V \right\}$ is the edge set, and $\boldsymbol{A}$ is the adjacency matrix. v_i and v_j denote the elements in node set. e_{ij} denote the element in edge set where the v_i and v_j are adjacent. The nodes and the edges can be associated with signals or features.

Definition II *Adjacency matrix*. The adjacency matrix

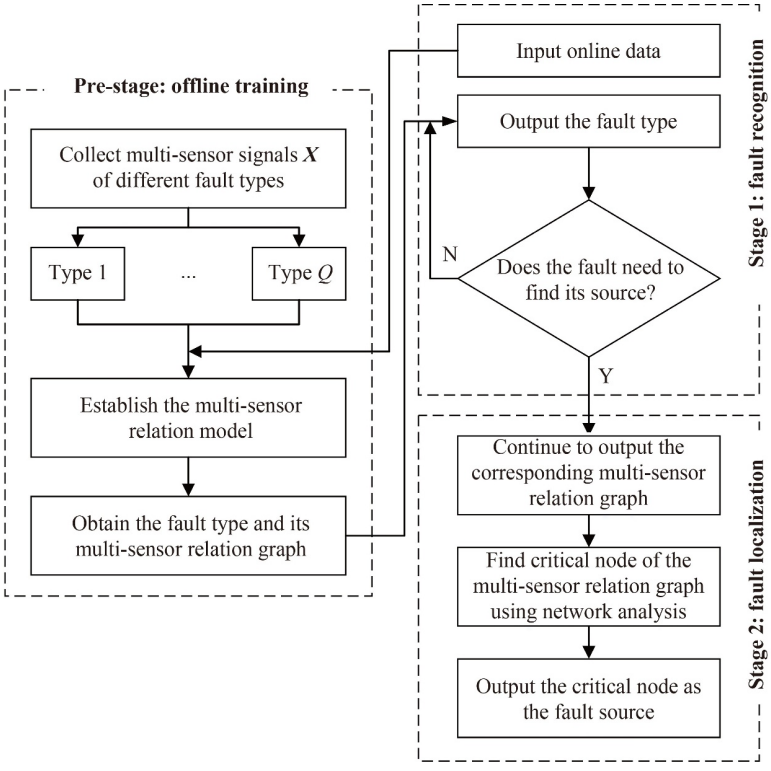

Fig. 1 Schematic diagram of the global fault diagnosis method.

is often denoted as $\boldsymbol{A} \in \Re^{N\times N}$, $A_{ij}=0$ if $(v_i,v_j)\notin E$ and $A_{ij}>0$ if $(v_i,v_j)\in E$, where N is the number of nodes, A_{ij} denotes the entry in the ith row and jth column of the adjacency matrix $\boldsymbol{A}$.

2.2 MSR model

As illustrated in Fig. 1, the MSR model is capable of recognizing fault types and modeling the corresponding MSRs simultaneously, which is quite different from directly using GNNs. The existing GNNs need pre-defined graphs as inputs, often calculated by similarities or dissimilarities from the multi-sensor signals. The MSR model tries to model and learn from the MSRs in a whole process, which avoids the trouble of pre-defined graph calculation. The MSR model contains graph structure learning and dynamic graph learning.

2.2.1 Graph structure learning

Graph structure learning aims at learning the explicit graph structure for MSRs. Two steps are conducted to model the explicit graph structure. The first step is to build a latent edge-type encoder, which can obtain the probability distribution of the latent edge type. The second step is to get the real edge type by sampling from the probability distribution of the latent edge type.

(1) Building latent edge type encoder

The goal of building the latent edge type encoder is to infer the distribution of the latent edge type. A transition matrix that can transform the multi-sensor signals to the edge domain must be determined to obtain the latent edge type. The formula $e_{ij}=(v_i,v_j)$ depicts the relationship between an edge and its endpoints in a graph. Edges can thus be represented by node combinations in sequence from the stariting point v_i to terminal point v_j. As shown in Fig. 2, the four sensors are modeled by nodes from v_1 to v_4. After 'from vertex to edge', sensor relations are modeled by edges from e_{12} to e_{34}. As a result, the transition matrix can be obtained by one-hot encoding to the endpoints of the edges, which allows multi-sensor signals to transform from the vertex domain to the edge domain and vice versa.

For the unknown graph structure with four sensor nodes in Fig. 2, the transition matrix of the two endpoints can be written as follows:

$$\boldsymbol{S}_{VI}=\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&0\\0&0&0&0&0&1\\0&0&0&0&0&0\end{bmatrix}^{\mathrm{T}},$$
$$\boldsymbol{S}_{VJ}=\begin{bmatrix}0&0&0&0&0&0\\1&0&0&0&0&0\\0&1&0&1&0&0\\0&0&1&0&1&1\end{bmatrix}^{\mathrm{T}}, \quad (1)$$

where $\boldsymbol{S}_{VI}$ is the transition matrix encoded from the endpoints in node set VI and $\boldsymbol{S}_{VJ}$ denotes from the endpoints in node set VJ. VI is the node set which

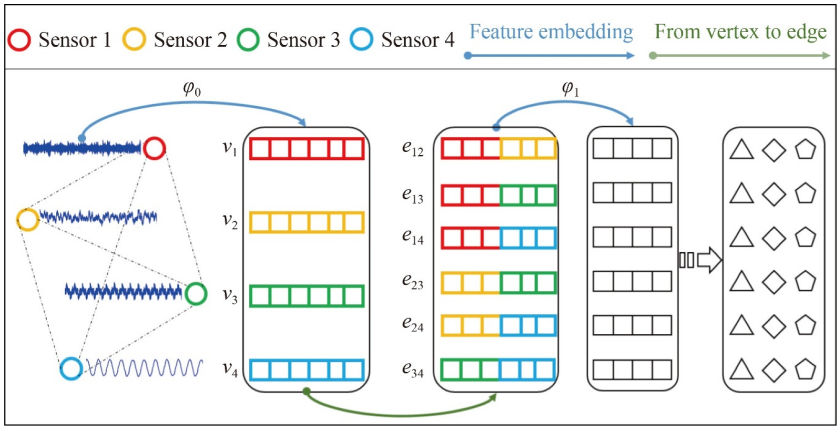

Fig. 2 Latent edge type encoder.

contains the starting points of the edges and VJ is the node set which contains the terminal points of the edges. The multi-layer perception is employed as the basic unit of the latent edge type encoder for feature embedding because of its simplicity and efficiency. Given the multi-sensor signals $\boldsymbol{X} = \{\boldsymbol{x}_m | m = 1, 2, ..., M\}$, where M denotes the number of the sensors and $\boldsymbol{x}_m$ denotes the time series collected from one of the M sensors, the signals are first mapped into the vertex domain to get a more compressed node feature $\boldsymbol{F}_0^{\mathrm{v}}$, which is formulated as follows:

$$\boldsymbol{F}_0^{\mathrm{v}} = \varphi_0(\boldsymbol{X}), \tag{2}$$

where $\varphi_0(\cdot)$ denotes the basic unit for the initial feature embedding in the vertex domain. Then the node feature is transformed to the edge domain, and thus the corresponding initial edge feature $\boldsymbol{F}_0^{\mathrm{e}}$ can be obtained:

$$\boldsymbol{F}_0^{\mathrm{e}} = \underset{\boldsymbol{S}\in\{\boldsymbol{S}_{VI},\boldsymbol{S}_{VJ}\}}{\mathrm{concat}}(\boldsymbol{S} \times \boldsymbol{F}_0^{\mathrm{v}}), \tag{3}$$

where $\boldsymbol{S}$ denotes the transition matrix and concat$(\cdot)$ denotes a concatenation operator.

The edge feature $\boldsymbol{F}_0^{\mathrm{e}}$ can be further embedded in the edge domain by the following:

$$\boldsymbol{F}_1^{\mathrm{e}} = \varphi_1(\boldsymbol{F}_0^{\mathrm{e}}), \tag{4}$$

where $\varphi_1(\cdot)$ denotes the multi-layer perception for feature embedding in the edge domain and $\boldsymbol{F}_1^{\mathrm{e}}$ is the features after the first time feature embedding in edge domain.

By repeating the above process, a deep feature embedding $\boldsymbol{F}^{\mathrm{e}}$ (i.e., the latent edge type) can be obtained. Finally, the probability distribution can be obtained by performing the softmax to the latent edge type. To obtain the updated node feature $\boldsymbol{F}^{\mathrm{v}}$, the inverse transformation from the edge domain to the vertex domain can be written as follows:

$$\boldsymbol{F}^{\mathrm{v}} = \boldsymbol{S}^{\mathrm{T}} \times \boldsymbol{F}^{\mathrm{e}}, \tag{5}$$

where $\boldsymbol{S} \in \{\boldsymbol{S}_{VI}, \boldsymbol{S}_{VJ}\}$.

(2) Sampling from latent edge type

Gumbel softmax [25] is adopted to obtain the real edge type from the latent edge type encoder. Gumbel sampling from the latent edge type is something like the occupancy of the space as far as possible, which helps to get the generalized edge type. Considering the latent edge type obtained by the encoder, the real edge type $\boldsymbol{Z}$ can be obtained by the following:

$$\boldsymbol{Z} = \mathrm{softmax}\left(\frac{\log \boldsymbol{\pi} + \boldsymbol{g}}{\tau}\right), \tag{6}$$

where $\boldsymbol{Z} = \{\boldsymbol{z}_k | k = 1, 2, ..., K\}$, K is the number of the edge type, $\boldsymbol{\pi}$ denotes the probability distribution of the latent edge type, $\boldsymbol{g} = \log(-\log(\boldsymbol{u}))$, $\boldsymbol{u} \sim \mathrm{uniform}(0, 1)$, and τ is a temperature parameter. As τ approaches 0, the distribution of $\boldsymbol{Z}$ converges to one-hot; otherwise, it tends to be smooth.

2.2.2 Dynamic graph learning

Dynamic graph learning aims to obtain the fault types and the MSR graphs. As illustrated in Fig. 3, the real edge type obtained through graph structure learning allows the sensor node signals to propagate across the self-learned graphs. The signals are first aggregated and updated at these sensor nodes. Then, the updated node features (i.e., messages) are assembled for dimension reduction to get fault labels. An attention mechanism is introduced to determine the best graph structure of the MSRs. A high attention score indicates that the corresponding structure carries the precise and necessary message for recognizing fault types. As a result, the informative graph structure is chosen as the MSRs to aid further fault localization. It is reasonable to localize fault sources with the MSR graphs because they are learned under the supervision of fault types. Dynamic graph learning will be elaborated on in the following steps.

(1) Node feature embedding

To obtain node feature embeddings from multiple sensors, the convolutional operation is employed to capture the features of temporal signals. For a one dimensional (1D) signal $\boldsymbol{x}_m$, the convolution is defined as follows:

$$y(d) = \sum_{r=0}^{R} w(r) x(d - r), \tag{7}$$

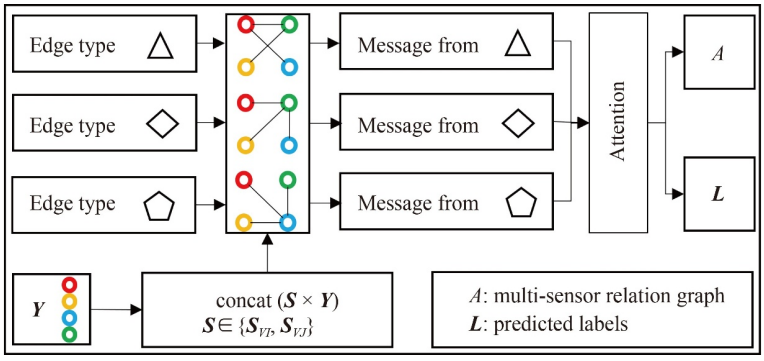

Fig. 3 Dynamic graph learning with the learned edge types.

where $w(r)$ is the rth element of a convolution kernel, R is the kernel size, $x(d-r)$ denotes the $(d-r)$th data point of the signals and $y(d)$ denotes the dth data point of the signals after convolution. In Fig. 3, the multi-sensor signals after node feature embedding are denoted as $\boldsymbol{Y}$.

(2) Message passing

Message passing is a generalized framework of the GNNs [26]. It is similar to a heat diffusion process in a graph. The temporal features $\boldsymbol{Y}$, which are learned in the vertex domain, are transformed to the edge domain with the transition matrices by the following:

$$\boldsymbol{Y}^{\mathrm{e}} = \underset{\boldsymbol{S}\in\{\boldsymbol{S}_{VI},\boldsymbol{S}_{VJ}\}}{\mathrm{concat}}(\boldsymbol{S}\times\boldsymbol{Y}), \tag{8}$$

where $\boldsymbol{Y}^{\mathrm{e}}$ are the transformed features in the edge domain. After the message passes across the whole graph $\boldsymbol{z}_k$, the aggregated message is formulated as follows:

$$\boldsymbol{h}_k = \sigma(\boldsymbol{z}_k \times \boldsymbol{Y}^{\mathrm{e}}), \tag{9}$$

where $\sigma(\cdot)$ is the non-linear activation function.

(3) Edge-type refinement

Attention score is calculated to determine the unique graph structure which contains the most informative message. For the message $\boldsymbol{h}_k \in \Re^{D\times 1}$, its attention score is calculated with the following:

$$a_k = \boldsymbol{p}^{\mathrm{T}} \times \sigma(\boldsymbol{w}\times\boldsymbol{h}_k + \boldsymbol{b}), \tag{10}$$

where $\boldsymbol{p}\in\Re^{D'\times 1}$ are learnable weights, $\boldsymbol{w}\in\Re^{D'\times D}$ and $\boldsymbol{b}\in\Re^{D'\times 1}$ denote the weights and bias of the non-linear transformation. A normalized learnable attention score can be obtained by the following:

$$\alpha_k = \mathrm{softmax}(a_k) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}. \tag{11}$$

In summary, the MSR model for fault diagnosis of machines can be represented as the Algorithm.

Algorithm MSR model

Input: $\boldsymbol{X}$, K
Output: $\boldsymbol{L}$, A
1: The edge type encoder encodes $\boldsymbol{X}$ into the latent edge type
$\boldsymbol{F}^{\mathrm{e}} \leftarrow \mathrm{encoder}(\boldsymbol{X})$
2: The real edge types are obtained through Gumbel softmax
$\boldsymbol{Z} \leftarrow \mathrm{gumbel\_softmax}(\boldsymbol{F}^{\mathrm{e}})$
3: The node signals are embedded into features
$\boldsymbol{Y} \leftarrow \mathrm{conv}(\boldsymbol{X})$
4: Message passing via the real edge types
Calculate $\boldsymbol{Y}^{\mathrm{e}}$ using Eq. (8)
For $\boldsymbol{z}_k \in \boldsymbol{Z}$,
calculate $\boldsymbol{h}_k$ using Eq. (9)
calculate α_k for $\boldsymbol{h}_k$ using Eq. (11)
update $\boldsymbol{h}_k$ using the normalized attention score by Hadamard product $\boldsymbol{h}_k \leftarrow \alpha_k \odot \boldsymbol{h}_k$
End For
Initialize $\boldsymbol{h} = 0$, $i = 0$, $\alpha = 0$
For $k = 1,2,...,K$,
$\boldsymbol{h} \leftarrow \boldsymbol{h} + \boldsymbol{h}_k$
If $\alpha_k \geqslant \alpha$,
$\alpha \leftarrow \alpha_k$
$i \leftarrow k$
End For
5: The informative edge type is $\boldsymbol{z}_i$ and its adjacency matrix $\boldsymbol{A}$ can be obtained using $\boldsymbol{z}_i$, $\boldsymbol{S}_{VI}$, $\boldsymbol{S}_{VJ}$
6: The predicted label is calculated by
$\boldsymbol{L} \leftarrow \mathrm{mlp}(\boldsymbol{h})$
return $\boldsymbol{L}$, A

3 Experimental verifications

3.1 Case study I: experiments on an induction motor

In this case, experiments were carried out on an induction motor. The experimental bench is shown in Fig. 4, which is composed of the test induction motor, an external bearing, a rotary encoder, and an eddy current brake. Four health conditions are considered for the test induction motor: normal condition (NC) and three typical electromechanical fault types. The three electromechanical faults include a broken rotor bar (BRB), an unbalanced voltage supply (UVS), and an open phase fault (OPF).

In the experimental investigations, the experiments on fault type recognition are conducted, and the experiments on fault source localization are designed. For the unbalanced voltage supply fault and the open phase fault, their fault sources can be found in Table 1. The sensor placement is shown in Fig. 4, current clamps and accelerometer sensors are employed for monitoring the motor condition. Two triaxle accelerometers are placed at the driven end and fan end of the motor. As shown in Fig. 4, the *x*-, *y*-, and *z*-axis of the accelerometer at the drive end are denoted as D*X*, D*Y*, and D*Z*, respectively. Similarly, F*X*, F*Y*, and F*Z* represent the three axes at the fan end. At the same time, the three phases of the motor are denoted as U, V, and W.

3.1.1 Results of fault recognition

The performance of the proposed method for recognizing fault types is presented in this section. Comparisons are conducted with the graph learning models that could learn from MSRs, including GCN [27], graph attention network (GAT) [28], simplifying graph convolutional network (SGC) [29], and high-order graph neural network (HGN) [30]. They need pre-defined graphs as inputs. The graph construction method should be decided first to construct pre-defined graphs with multi-sensor signals.

The full graph is the most intuitive one to construct a graph with the multi-sensor signals, whereas the *k*-nearest neighbor graph (i.e., *k*-NN graph) [4] and the ε-neighbourhood graph [5] are more popular in existing multi-senor-based methods. We denote *k*-NN graphs with weighted edges as *k*-NN-w and *k*-NN graphs with binary edges as *k*-NN-b. Similarly, full graphs are denoted as Full-w and Full-b. The ε-neighbourhood graphs are denoted as ε-w and ε-b. However, learning from MSRs is a graph classification problem. Given diverse graph pooling techniques, the architectures of graph learning models for graph classification vary differently. For a fair comparison, the architecture adopted in the comparisons remains the same for all the graph learning models [26]. Meanwhile, the experimental setting for training the graph learning models is the same as the MSR model.

The precision, recall and F1-score of the MSR model are reported in Table 2, and the average classification accuracy of the MSR model reaches 0.9980. Tables 3–5 show the results of the graph learning models (i.e., GCN, GAT, SGC, and HGN) with pre-defined graphs whose similarity metrics are calculated by Pearson correlation, Euclidean distance, and cosine similarity, respectively. As shown in Table 3, GCN and SGC perform well when using a full graph, but their performance varies vastly with different graph structures. GAT shows the worst

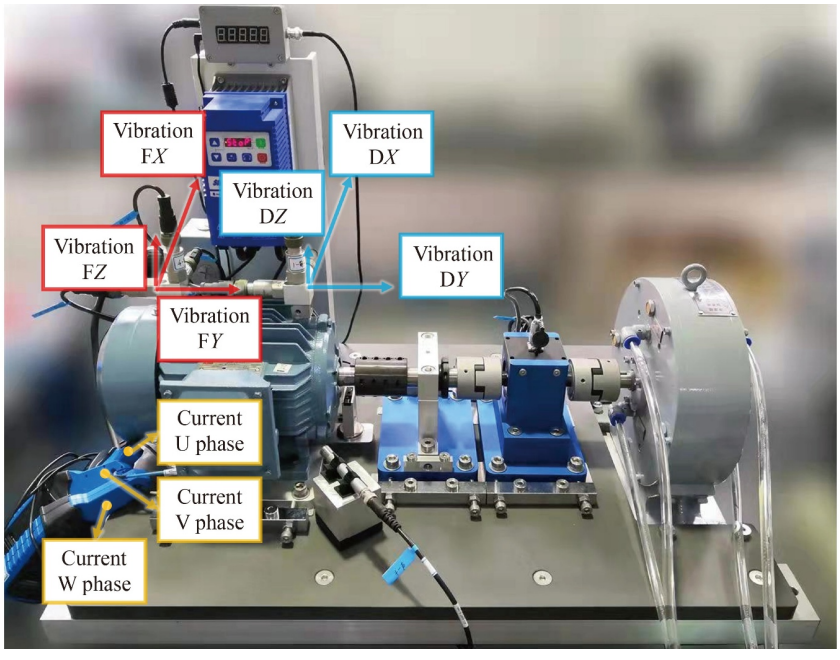

Fig. 4 Experimental bench of an induction motor.

Table 1 Multi-class faults and the sources

Fault source notation	NC	BRB	UVS	OPF
D*X*, D*Y*, D*Z*	–	–	×	×
F*X*, F*Y*, F*Z*	–	–	×	×
U	–	–	×	×
V	–	–	×	√
W	–	–	√	×

Notes: '–' denotes no source for the fault type; '√' denotes the source is 'True' for the fault type in the experiment and '×' denotes 'False'.

Table 2 Evaluation metrics of fault recognition

Fault type	Evaluation metric		
	Precision	Recall	F1-score
NC	0.9924	1.0000	0.9962
BRB	1.0000	1.0000	1.0000
UVS	1.0000	1.0000	1.0000
OPF	1.0000	0.9924	0.9962

Table 3 Case I: average classification accuracy with Pearson correlation

Graph learning model	Average classification accuracy					
	k-NN-w	*k*-NN-b	*ε*-w	*ε*-b	Full-w	Full-b
GCN	0.8314	0.7955	0.9362	0.9618	0.9967	0.9720
GAT	0.8821	0.7935	0.7845	0.8030	0.6998	0.8645
SGC	0.8437	0.8219	0.9368	0.9811	0.9967	0.9908
HGN	0.9967	0.9986	0.9921	0.9967	0.9986	0.9980

Table 4 Case I: average classification accuracy with Euclidean distance

Graph learning model	Average classification accuracy					
	k-NN-w	*k*-NN-b	*ε*-w	*ε*-b	Full-w	Full-b
GCN	0.9015	0.9433	0.9962	0.9943	0.9962	0.9641
GAT	0.8910	0.8683	0.7301	0.8001	0.8390	0.8702
SGC	0.9460	0.9524	0.9839	0.9752	0.9867	0.9550
HGN	0.9829	0.9947	0.9981	0.9986	0.9986	0.9967

Table 5 Case I: average classification accuracy with Cosine similarity

Graph learning model	Average classification accuracy					
	k-NN-w	*k*-NN-b	*ε*-w	*ε*-b	Full-w	Full-b
GCN	0.8503	0.8390	0.9388	0.9713	0.9934	0.9824
GAT	0.9507	0.9427	0.7194	0.7310	0.8125	0.7821
SGC	0.8248	0.8381	0.9394	0.9824	0.9941	0.9772
HGN	0.9980	0.9980	0.9869	0.9967	0.9986	0.9980

performance in classification accuracy, whereas HGN is superior to other models. Furthermore, HGN shows consistent performance across all graph construction methods. In the comparisons, the MSR model achieves promising results in terms of classification accuracy. Similar results are shown in Tables 4 and 5 using Euclidean distance and Cosine similarity. From the above analysis, most existing graph learning models are sensitive to graph structures; therefore, the accuracies are influenced. With the dynamic graph learning process, the MSR model could effectively use self-learned graphs instead of pre-defined graphs to avoid this problem.

In the case study, comparisons are also conducted with the traditional methods for multi-sensor-based fault diagnosis. Detailed information for the traditional methods is provided in Table 6. The proposed method achieves competitive accuracy with these methods.

3.1.2 Results of fault localization

The performance of the proposed method in localizing fault sources is further analyzed in this section. As stated earlier, fault localization aims to locate the faulty sources for the motor under certain types of health conditions. For the faults of UVS and OPF, as listed in Table 1, the unbalanced phase is set as the W-phase, and the open phase is set as the V-phase in the experiment. Graph analytics for the learned MSRs is conducted to locate the faulty phase of the corresponding faults. Centrality measures are vital tools for understanding the importance of the nodes in a graph. Since graphs are abstract models for depicting relations in various fields, researchers have studied different centrality measures that identify critical nodes to meet diverse goals [24]. The degree and eigenvector centrality are analyzed for fault localization in the following sub-sections.

(1) Centrality measures

Degree centrality calculates the importance of a node according to the number of its direct links. It is the most straightforward measure because it consists only of one-hop connections. However, it is still quite effective in finding popular nodes, very connected nodes, and nodes that are likely to hold the most information. For an undirected graph, its formulation is as follows:

$$C_{\mathrm{D}}(i) = \sum\nolimits_j A_{ij}, \tag{12}$$

where $C_{\mathrm{D}}(i)$ denotes the degree centrality of the ith node.

Eigenvector centrality is an extension of the degree

Table 6 Comparisons with traditional multi-sensor fault diagnosis methods [14,15,17]

Input	Model	Publication	Category of the method	Accuracy
FFT + 2D matrix	2D convolutional neural network	Azamfar et al. [14]	Data-level fusion	0.9728
Statistical features	Covariance matrix fusion with multi-kernel learning	Li et al. [15]	Feature-level fusion	0.9540
Raw signal + stacked wavelet auto-encoder	Improved auto-encoder with voting strategy	Shao et al. [17]	Decision-level fusion	0.9968

centrality. It considers the number of links held by both the current node and its connected ones. This measure shows an all-around ability to identify influential nodes, especially for understanding how features propagate across the whole graph. Therefore, it is also used for analyzing MSRs. For an undirected graph, it can be calculated by the following:

$$C_{\mathrm{E}}(i) = \lambda^{-1} \sum_j A_{ij} C_{\mathrm{E}}(j), \tag{13}$$

where $C_{\mathrm{E}}(i)$ denotes the eigenvector centrality of the ith node and λ is the largest eigenvalue of $\boldsymbol{A}$.

(2) MSR analysis

Figure 5 exhibits the MSR graphs learned by the MSR model. The circle is uniformly occupied by sensor nodes. The red lines inside the circle are edges, which represent messages passing between different node pairs. As depicted in Fig. 1, the MSR model provides simultaneous outputs of the fault types and the MSR graphs. Once the fault type is recognized, its MSR graphs can be used to understand how the model distinguishes one health condition of the motor from others.

For the motor in NC or with the fault BRB, as shown in Figs. 5(a) and 5(b), the dynamic graph learning prompts message passing among the fan end vibrations and driven end vibrations. It indicates that only vibration data are adequate for the model to distinguish these two types. In the case study, the simulated BRB fault is the same as the mechanical rotor imbalance. Hence, it is not hard to find that the fault symptom of the BRB fault can be reflected by vibrations. For the motor with the faults UVS and OPF, the MSR graphs in Figs. 5(c) and 5(d) are different from those in Figs. 5(a) and 5(b). Therefore, the vibration data are insufficient for the model to distinguish the two types from NC and BRB. As shown in Figs. 5(c) and 5(d), additional information from current signals is included to recognize the fault UVS or OPF, demonstrating that the fault symptoms of UVS and OPF are reflected by the current signals against NC and BRB when using the model for fault recognition.

The degree centrality and the eigenvector centrality are calculated for the MSR graphs to localize the fault sources of UVS and OPF. The results are presented in Figs. 6 and 7. Sources are represented by the node in the circle. The color bar shows the value range of the centrality measures.

As shown in Fig. 6, the bigger node in the circle is identified as the critical node since it has the maximum degree of centrality. For the fault UVS, the critical node is identified as node W in Fig. 6(c), consistent with the experimental setting in Table 1. The node V identified in Fig. 6(d) is exactly the open phase settled for the fault OPF. For the fault BRB, it is reasonable to see the critical

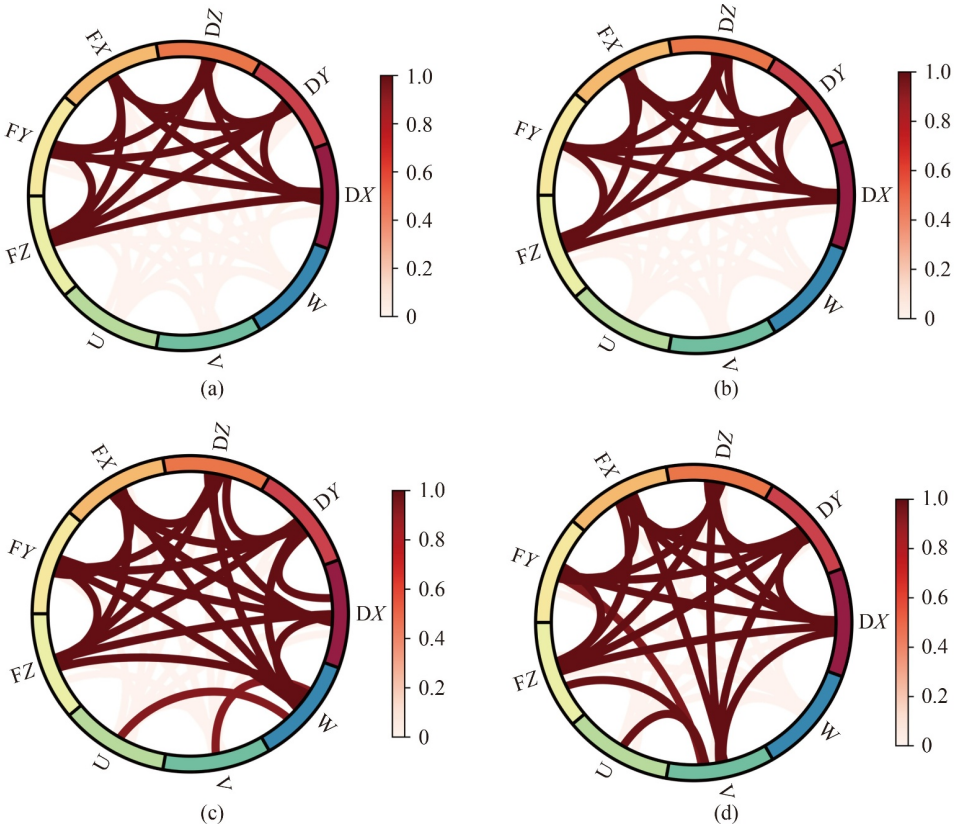

Fig. 5 Case I: multi-sensor relation graph of different health conditions: (a) normal condition, (b) broken rotor bar, (c) unbalanced voltage supply, and (d) open phase fault.

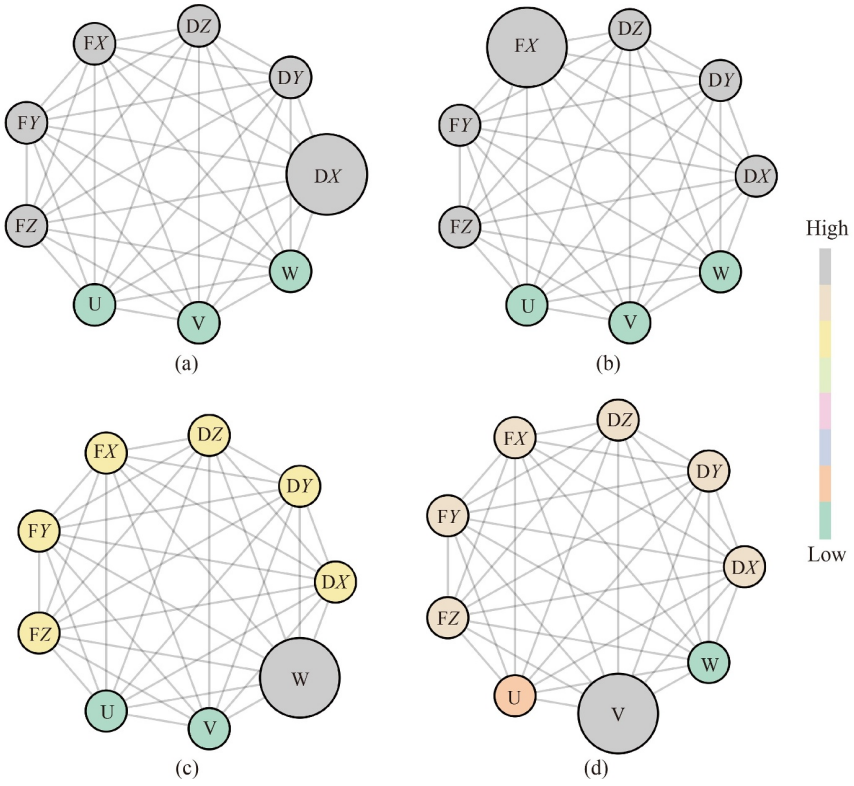

Fig. 6 Case I: degree centrality of different health conditions: (a) normal condition, (b) broken rotor bar, (c) unbalanced voltage supply, and (d) open phase fault.

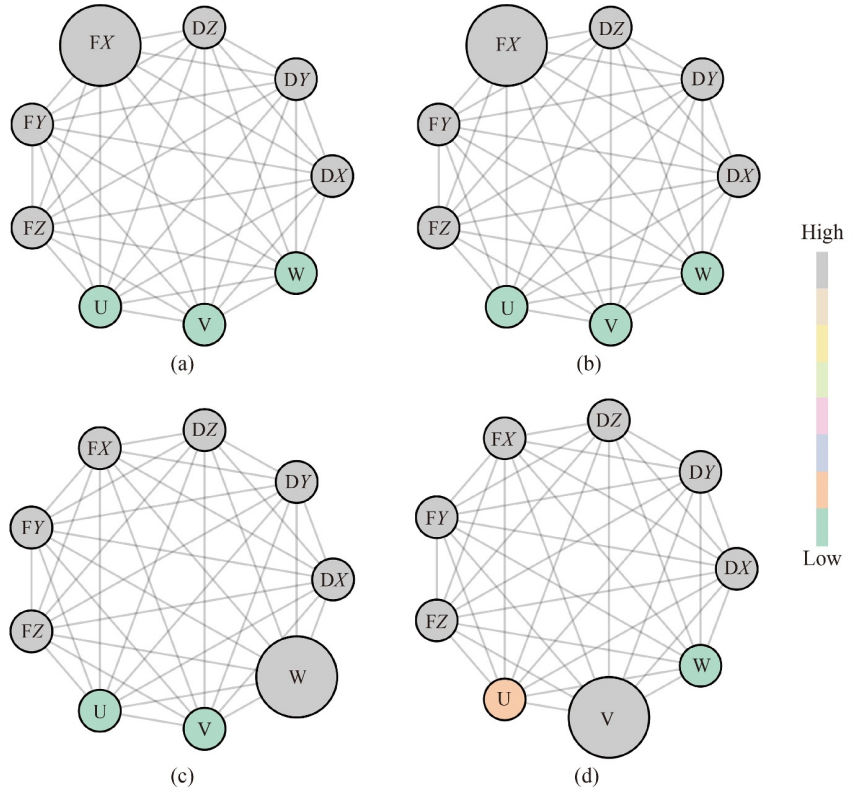

Fig. 7 Case I: eigenvector centrality of different health conditions: (a) normal condition, (b) broken rotor bar, (c) unbalanced voltage supply, and (d) open phase fault.

node comes from vibration sources because it has similar fault symptoms to the mechanical rotor imbalance. Similar findings can be seen in Fig. 7 with eigenvector centrality. Note that both the degree centrality and the eigenvector centrality reveal the F*X* as the critical node for the fault BRB. In the case study, when simulating the fault BRB, the defect is introduced close to the fan end at the rotor bar of the motor, showing the self-learned graph sensitivity.

3.2 Case study II: experiments on a centrifugal pump

In this case, the bearing faults (i.e., outer race fault, inner race fault, and ball fault) and a typical hydraulic fault (i.e., cavitation fault (CF)) are simulated. In the context of centrifugal pumps, cavitation implies a dynamic process of formation of bubbles inside the liquid, their growth, and subsequent collapse as the liquid flows through the pump. Figure 8 depicts the experimental bench of a centrifugal pump.

The centrifugal pump is installed on the fixed base of the experimental bench. It is connected to the water tank by a circulating pipe. The yellow arrow shows the direction of the water flow in the circulating pipe. The motor shaft drives the centrifugal pump. Water is pumped from the tank to the inlet pipe, passes through the centrifugal pump to the outlet pipe, and then goes back to the water tank. A manual valve is positioned at the inlet pipe to regulate the flow in the circulating pipe. Low flow-rate working conditions are reported to start the formation of vapor bubbles eventually, resulting in cavitation in centrifugal pumps [31]. Consequently, incipient cavitation is expected to appear in the experiment by modulating the manual valve. As shown in Fig. 8, three accelerometers were used to monitor vibration, one for the motor, one for the test bearing, and one for the centrifugal pump. The flowmeter measures the flow rate of water in the outlet pipe. The pressure sensor measures the pressure in the outlet pipe.

3.2.1 Results of fault recognition

In the experiment, five types of health conditions are included: NC, CF, inner race fault of bearing (IF), outer race fault of bearing (OF), and ball fault of bearing (BF). Comparisons are conducted with the same graph learning models and pre-defined graphs as in Section 3.1.1. The architecture of the graph learning models also remains the same. The average classification accuracy of the MSR model reaches 99.21%. The comparison results are listed in Tables 7–9. In the case study, the MSR model shows the advantages of classifying fault types against the compared graph learning models. From Tables 7–9, the proper similarity metric is hard to determine, while a proper graph structure and a suitable graph learning model are required for a multi-sensor-based fault diagnosis with GNNs.

3.2.2 Results of fault localization

Figure 9 exhibits the MSR graphs learned of different health conditions in the case. Figures 10 and 11 provide the results of identifying the critical nodes using the degree centrality and the eigenvector centrality, respectively.

In Fig. 9, the node symbol N1 represents the sensor at the motor, the node symbol N2 represents the sensor at

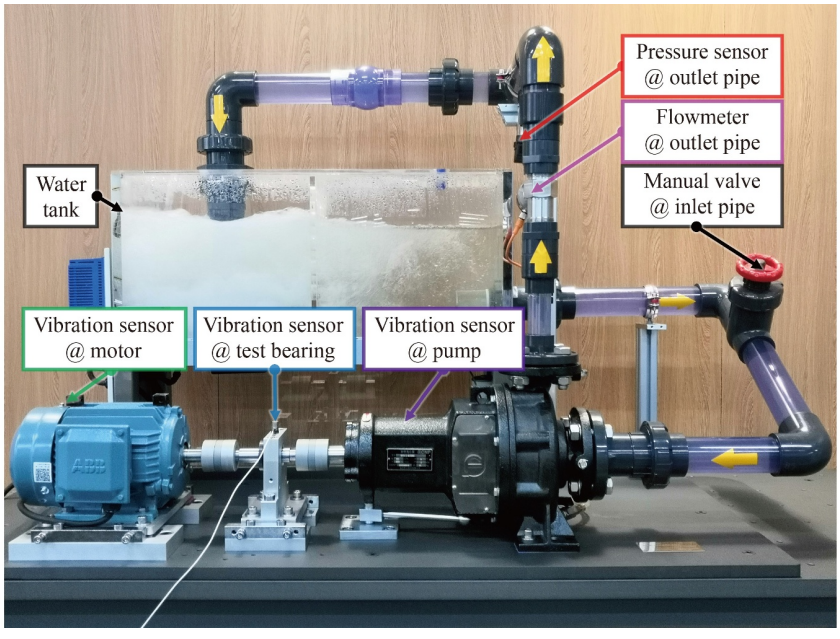

Fig. 8 Experimental bench of a centrifugal pump.

Table 7 Case II: average classification accuracy with Pearson correlation

Graph learning model	Average classification accuracy					
	k-NN-w	*k*-NN-b	*ε*-w	*ε*-b	Full-w	Full-b
GCN	0.8015	0.7453	0.8046	0.8265	0.7859	0.7656
GAT	0.6156	0.5625	0.6218	0.5812	0.5453	0.5765
SGC	0.7937	0.7703	0.8062	0.7562	0.8359	0.7765
HGN	0.7328	0.7765	0.7171	0.7828	0.7953	0.7218

Table 8 Case II: average classification accuracy with Euclidean distance

Graph learning model	Average classification accuracy					
	k-NN-w	*k*-NN-b	*ε*-w	*ε*-b	Full-w	Full-b
GCN	0.7546	0.8125	0.7656	0.7203	0.7984	0.7687
GAT	0.5406	0.5671	0.5719	0.5719	0.5672	0.5859
SGC	0.7906	0.8656	0.8094	0.8328	0.7968	0.8156
HGN	0.8563	0.8344	0.8734	0.7672	0.8047	0.8141

Table 9 Case II: average classification accuracy with Cosine similarity

Graph learning model	Average classification accuracy					
	k-NN-w	*k*-NN-b	*ε*-w	*ε*-b	Full-w	Full-b
GCN	0.8406	0.7828	0.7062	0.7797	0.7344	0.7797
GAT	0.5719	0.5625	0.5859	0.5813	0.5734	0.5531
SGC	0.7641	0.8391	0.7875	0.7828	0.7781	0.7922
HGN	0.7937	0.8656	0.8891	0.7578	0.7828	0.7922

the test bearing, the node symbol N3 represents the sensor at the pump, the node symbol N4 represents the flowmeter at the outlet pipe of the pump, and the node symbol N5 represents the pressure sensor at the outlet pipe of the pump. As can be seen in Figs. 9(a) and 9(c)–9(e), the MSR graphs of the four health conditions are similar, which demonstrates that these four types are distinguished by the MSR model using the information from the same sensors. The results for the four health conditions in Figs. 10 and 11 show that the critical node identified by the two centrality measures is the same.

Figure 9(b) shows the MSR graph for the CF. Cavitation is an undesirable phenomenon occurred in the operation of hydraulic pumps. It will damage mechanical components and deteriorate the performance of pumps. As pointed out in Ref. [31], a low flow rate working condition will result in cavitation in centrifugal pumps. As a result, CF is simulated by manipulating the valve opening in the case. Since the flow rate is the directly controlled variable in our experiment, this work treats the flow rate as the source of the CF. As shown in Fig. 9(b), the MSR model uses the flowmeter and pressure sensor information to distinguish the CF from others. The source of the CF needs to be identified to verify the proposed method. The critical node is denoted by a bigger one in the circle. As shown in Fig. 10(b), N4 is identified as the critical node, consistent with the experimental setting for cavitation. Similar results can be found in Fig. 11(b).

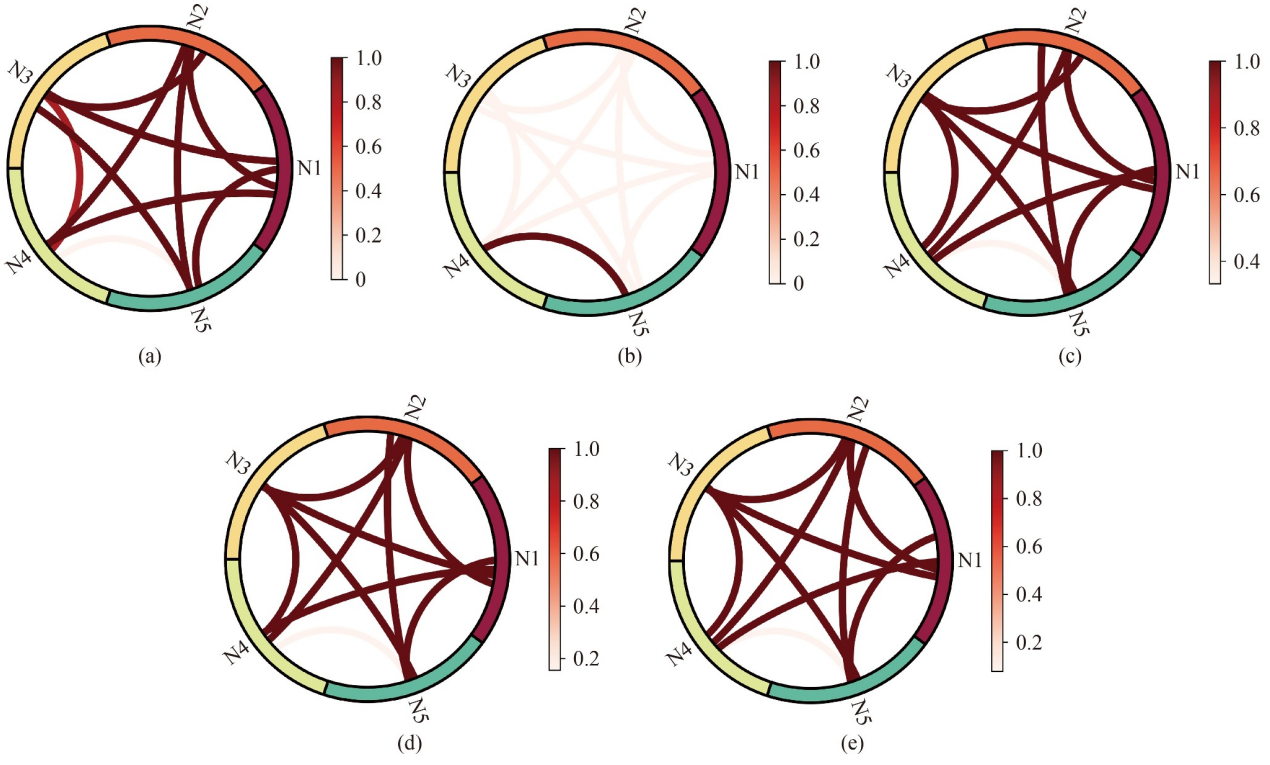

Fig. 9 Case II: multi-sensor relation graph of different health conditions: (a) normal condition, (b) cavitation fault, (c) inner race fault of bearing, (d) outer race fault of bearing, and (e) ball fault of bearing.

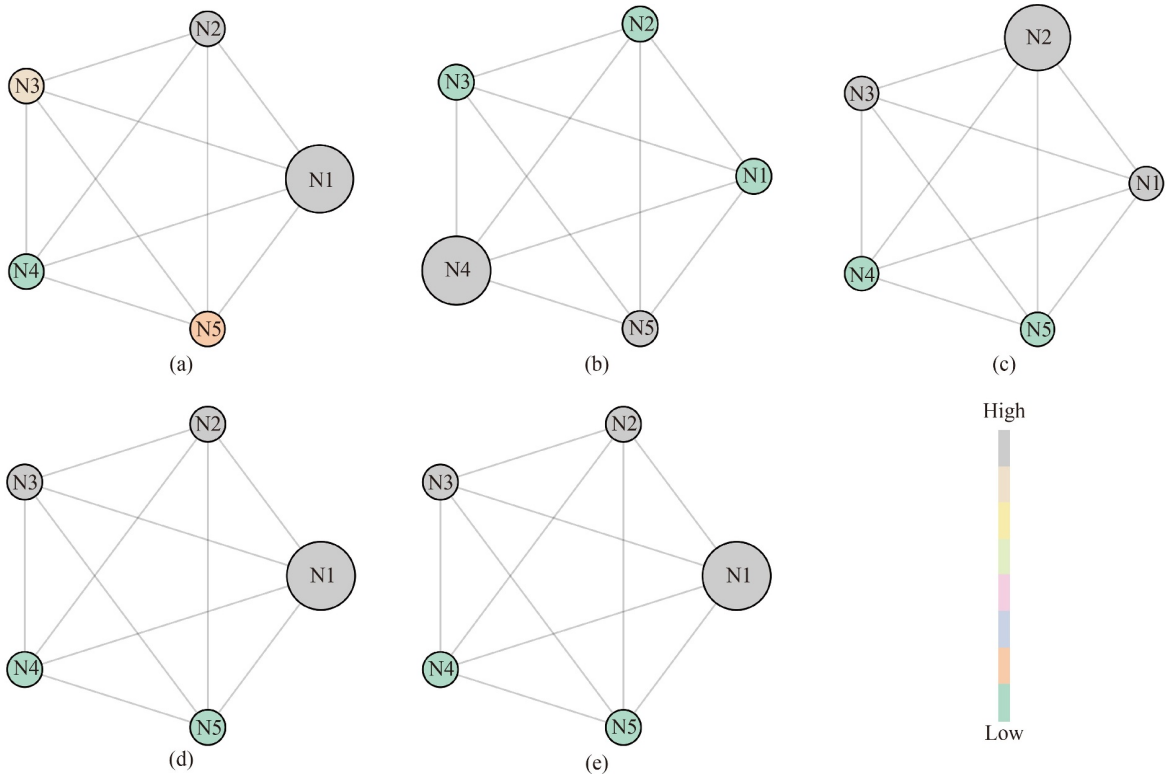

Fig. 10 Case II: degree centrality of different health conditions: (a) normal condition, (b) cavitation fault, (c) inner race fault of bearing, (d) outer race fault of bearing, and (e) ball fault of bearing.

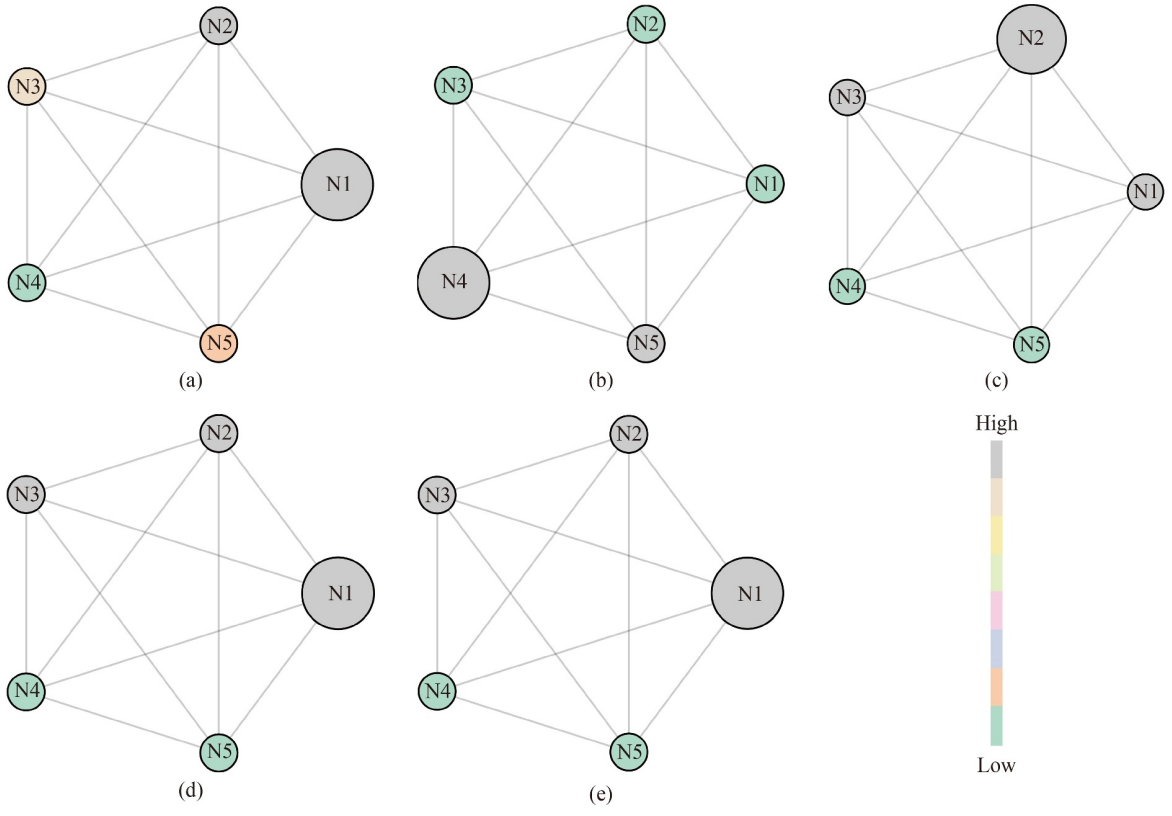

Fig. 11 Case II: eigenvector centrality of different health conditions: (a) normal condition, (b) cavitation fault, (c) inner race fault of bearing, (d) outer race fault of bearing, and (e) ball fault of bearing.

4 Conclusions

This article proposes a GFD method using MSRs and network analysis. The proposed method aims to recognize multi-class faults and localize fault sources. The key to the proposed method for fault localization is the MSR model. In the model, the MSR graphs can be learned through the dynamic graph learning process without requiring extra similarity metrics and graph construction methods. Meanwhile, the self-learned graphs help us understand how the model distinguishes between different fault types. Centrality measures are applied to the MSR graphs to identify fault sources. Multi-source faults are designed on two major machines (i.e., motor and pump) to validate the proposed method. Results show that the proposed method can localize the correct source. Besides, it reaches comparable or even better results for multi-class fault recognition.

Nomenclature

Abbreviations

1D	One dimensional
BF	Ball fault
BRB	Broken rotor bar
CF	Cavitation fault
Full-b	Full graph with binary edges
Full-w	Full graph with weighted edges
GAT	Graph attention network
GCN	Graph convolutional network
GFD	Global fault diagnosis
GNN	Graph neural network
HGN	High-order graph network
IF	Inner race fault of bearing
k-NN	*k*-nearest neighbor
k-NN-b	*k*-nearest neighbor graph with binary edges
k-NN-w	*k*-nearest neighbor graph with weighted edges
MSR	Multi-sensor relation
NC	Normal condition
OF	Outer race fault of bearing
OPF	Open phase fault
SGC	Simplifying graph convolutional network
UVS	Unbalanced voltage supply
ε-b	ε-neighbourhood graph with binary edges
ε-w	ε-neighbourhood graph with weighted edges

Variables

a_k	Attention score of $\boldsymbol{z}_k$
A	Adjacency matrix
A_{ij}	Elements in A
concat(·)	A concatenation operator
$C_{\mathrm{D}}(i)$	Degree centrality of the ith node
$C_{\mathrm{E}}(i)$	Eigenvector centrality of the ith node
e_{ij}	Edge connects v_i and v_j
$\boldsymbol{F}^{\mathrm{e}}$	Edge feature
$\boldsymbol{F}^{\mathrm{e}}_0$	Initial edge features
$\boldsymbol{F}^{\mathrm{e}}_1$	Edge features after the first time feature embedding in edge domain
$\boldsymbol{F}^{\mathrm{v}}$	Node feature
$\boldsymbol{F}^{\mathrm{v}}_0$	Initial node features
$\boldsymbol{g}$	Gumbel distribution
G	Graph
$\boldsymbol{h}_k \in \Re^{D\times 1}$	Message from $\boldsymbol{z}_k$, D is the dimension of $\boldsymbol{h}_k$
K	Number of the edge type
$\boldsymbol{L}$	Predicted label
M	Number of the sensors
N	Number of nodes
$\boldsymbol{p} \in \Re^{D'\times 1}$	Learnable weights
R	Kernel size
$\boldsymbol{S}$	Transition matrix
$\boldsymbol{S}_{VI}$, $\boldsymbol{S}_{VJ}$	Transition matrix encoded from the endpoints in node set VI and VJ, respectively
$\boldsymbol{u}$	Uniform distribution
v_i, v_j	Endpoints of e_{ij}
V	Node set
VI, VJ	Node sets of v_i and v_j, respectively
$w(r)$	rth element of a convolution kernel
$\boldsymbol{w} \in \Re^{D'\times D}$, $\boldsymbol{b} \in \Re^{D'\times 1}$	Weights and bias of the non-linear transformation, respectively
x	Element of $\boldsymbol{x}_m$
$\boldsymbol{x}_m$ (m = 1,2,…,M)	Time series collected from one of the M sensors
$\boldsymbol{X}$	Multi-sensor signals
$y(d)$	dth data point of signals after convolution
$\boldsymbol{Y}$	Temporal features
$\boldsymbol{Y}^{\mathrm{e}}$	Transformed features in the edge domain
$\boldsymbol{z}_k$	kth real edge type
$\boldsymbol{Z}$	Real edge type
α_k	Normalised attention score of $\boldsymbol{z}_k$
$\varphi_0(\cdot)$, $\varphi_1(\cdot)$	Basic unit of the latent edge type encoder for the initial and first feature embeddings in the vertex domain, respectively
$\sigma(\cdot)$	Non-linear activation function
τ	A temperature parameter
λ	Largest eigenvalue of $\boldsymbol{A}$
$\boldsymbol{\pi}$	Probability distribution of the latent edge type

Acknowledgements This study was supported by the National Natural Science Foundation of China (Grant No. 52025056) and the Fundamental Research Funds for the Central Universities.

References

1. Lei Y G. Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery. Oxford: Elsevier, 2017
2. Chen X F, Wang S B, Qiao B J, Chen Q. Basic research on machinery fault diagnostics: past, present, and future trends. Frontiers of Mechanical Engineering, 2018, 13(2): 264–291
3. Zheng P, Wang H H, Sang Z Q, Zhong R Y, Liu Y K, Liu C, Mubarok K, Yu S Q, Xu X. Smart manufacturing systems for Industry 4.0: conceptual framework, scenarios, and future perspectives. Frontiers of Mechanical Engineering, 2018, 13(2): 137–150
4. Zhou K B, Yang C Y, Liu J, Xu Q. Deep graph feature learning-based diagnosis approach for rotating machinery using multi-sensor data. Journal of Intelligent Manufacturing, 2023, 34(4): 1965–1974
5. Li T F, Zhao Z B, Sun C, Yan R Q, Chen X F. Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction. Reliability Engineering & System Safety, 2021, 215: 107878
6. Lei Y G, Jia F, Lin J, Xing S B, Ding S X. An intelligent fault diagnosis method using unsupervised feature learning towards mechanical Big Data. IEEE Transactions on Industrial Electronics, 2016, 63(5): 3137–3147
7. Wang S H, Xiang J W, Zhong Y T, Tang H S. A data indicator-based deep belief networks to detect multiple faults in axial piston pumps. Mechanical Systems and Signal Processing, 2018, 112: 154–170
8. Wang S H, Xiang J W, Zhong Y T, Zhou Y Q. Convolutional neural network-based hidden Markov models for rolling element bearing fault identification. Knowledge-Based Systems, 2018, 144: 65–76
9. Wang S H, Xiang J W. A minimum entropy deconvolution-enhanced convolutional neural networks for fault diagnosis of axial piston pumps. Soft Computing, 2020, 24(4): 2983–2997
10. Zhang Y, Lu W X, Chu F L. Planet gear fault localization for wind turbine gearbox using acoustic emission signals. Renewable Energy, 2017, 109: 449–460
11. Hajary A, Kianinezhad R, Seifossadat S G, Mortazavi S S, Saffarian A. Detection and localization of open-phase fault in three-phase induction motor drives using second order rotational park transformation. IEEE Transactions on Power Electronics, 2019, 34(11): 11241–11252
12. Haje Obeid N, Battiston A, Boileau T, Nahid-Mobarakeh B. Early intermittent inter-turn fault detection and localization for a permanent magnet synchronous motor of electrical vehicles using wavelet transform. IEEE Transactions on Transportation Electrification, 2017, 3(3): 694–702
13. Lei Y G, Yang B, Jiang X W, Jia F, Li N P, Nandi A K. Applications of machine learning to machine fault diagnosis: a review and roadmap. Mechanical Systems and Signal Processing, 2020, 138: 106587
14. Azamfar M, Singh J, Bravo-Imaz I, Lee J. Multisensor data fusion for gearbox fault diagnosis using 2-D convolutional neural network and motor current signature analysis. Mechanical Systems and Signal Processing, 2020, 144: 106861
15. Li X, Zhong X, Shao H D, Han T, Shen C Q. Multi-sensor gearbox fault diagnosis by using feature-fusion covariance matrix and multi-Riemannian kernel ridge regression. Reliability Engineering & System Safety, 2021, 216: 108018
16. Wang Y X, Liu F, Zhu A H. Bearing fault diagnosis based on a hybrid classifier ensemble approach and the improved Dempster−Shafer theory. Sensors, 2019, 19(9): 2097
17. Shao H D, Lin J, Zhang L W, Galar D, Kumar U. A novel approach of multisensory fusion to collaborative fault diagnosis in maintenance. Information Fusion, 2021, 74: 65–76
18. Niu G X, Liu E H, Wang X, Ziehl P, Zhang B. Enhanced discriminate feature learning deep residual CNN for multi-task bearing fault diagnosis with information fusion. IEEE Transactions on Industrial Informatics, 2023, 19(1): 762–770
19. Liang M X, Zhou K. A hierarchical deep learning framework for combined rolling bearing fault localization and identification with data fusion. Journal of Vibration and Control, 2022 (in press)
20. Wu F, Zhao J. Current similarity analysis-based open-circuit fault diagnosis for two-level three-phase PWM rectifier. IEEE Transactions on Power Electronics, 2017, 32(5): 3935–3945
21. Irhoumah M, Pusca R, Lefèvre E, Mercier D, Romary R. Information fusion with correlation coefficient for detecting inter-turn short circuit faults in asynchronous machines. In: Proceedings of 2019 IEEE the 12th International Symposium on Diagnostics for Electrical Machines, Power Electronics, and Drives (SDEMPED). Toulouse: IEEE, 2019, 232–237
22. Kong L, Nian H. Fault detection and location method for mesh-type dc microgrid using Pearson correlation coefficient. IEEE Transactions on Power Delivery, 2021, 36(3): 1428–1439
23. Borsboom D, Deserno M K, Rhemtulla M, Epskamp S, Fried E I, McNally R J, Robinaugh D J, Perugini M, Dalege J, Contantini G, Isvoranu A M, Wysocki A C, van Borkulo C D, van Bork R, Waldorp L J. Network analysis of multivariate data in psychological science. Nature Reviews Methods Primers, 2021, 1(1): 58
24. Segarra S, Ribeiro A. Stability and continuity of centrality measures in weighted graphs. IEEE Transactions on Signal Processing, 2016, 64(3): 543–555
25. Kipf T, Fetaya E, Wang K C, Welling M, Zemel R. Neural relational inference for interacting systems. In: Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018, 2688–2697
26. Fey M, Lenssen J E. Fast graph representation learning with PyTorch Geometric. arXiv Preprint, 2019, arXiv: 1903.02428
27. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. arXiv Preprint, 2016, arXiv: 1609.02907
28. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. arXiv Preprint, 2017, arXiv: 1710.10903
29. Wu F, de Souza A H, Zhang T Y, Fifty C, Yu T, Weinberger K Q.

Simplifying graph convolutional networks. Proceedings of Machine Learning Research, 2019, 1525919

30. Morris C, Ritzert M, Fey M, Hamilton W L, Lenssen J E, Rattan G, Grohem M. Weisfeiler and leman go neural: higher-order graph neural networks. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 4602–4609
31. Tabar M T S, Majidi S H, Poursharifi Z. Investigation of recirculation effects on the formation of vapor bubbles in centrifugal pump blades. International Journal of Mechanical and Mechatronics Engineering, 2011, 5(1): 80–85