Interaction-Aware Graph Neural Networks for Fault Diagnosis of Complex Industrial Processes

Dongyue Chen, Ruonan Liu, *Member, IEEE*, Qinghua Hu, *Senior Member, IEEE*, and Steven X. Ding

***Abstract*— Fault diagnosis of complex industrial processes becomes a challenging task due to various fault patterns in sensor signals and complex interactions between different units. However, how to explore the interactions and integrate with sensor signals remains an open question. Considering that the sensor signals and their interactions in an industrial process with the form of nodes and edges can be represented as a graph, this article proposes a novel interaction-aware and data fusion method for fault diagnosis of complex industrial processes, named interaction-aware graph neural networks (IAGNNs). First, to describe the complex interactions in an industrial process, the sensor signals are transformed into a heterogeneous graph with multiple edge types, and the edge weights are learned by the attention mechanism, adaptively. Then, multiple independent graph neural network (GNN) blocks are employed to extract the fault feature for each subgraph with one edge type. Finally, each subgraph feature is concatenated or fused by a weighted summation function to generate the final graph embedding. Therefore, the proposed method can learn multiple interactions between sensor signals and extract the fault feature from each subgraph by message passing operation of GNNs. The final fault feature contains the information from raw data and implicit interactions between sensor signals. The experimental results on the three-phase flow facility and power system (PS) demonstrate the reliable and superior performance of the proposed method for fault diagnosis of complex industrial processes.**

***Index Terms*— Complex industrial processes, fault diagnosis, fault feature extraction, graph neural network (GNN), interaction-aware.**

I. Introduction

With the rapid development of complex industrial processes, such as power systems (PSs) and chemical processes, an effective and reliable health monitoring system has been increasingly necessary to reduce maintenance costs and guarantee the safety of industrial systems. In general, a monitoring system includes numerous sensors and devices for gathering multiple sensor measurements. These multivariate sensor readings are characterized by high-dimensional and complicated interactions due to the large-scale and complex control engineering of the process industries [1]. The monitoring system also provides maintenance records, which makes it possible to employ a supervised machine learning method. Thus, fault diagnosis of the complex industrial processes could be defined as a recognition task based on multivariate time-series signals, which has received more and more attention in both research and industry [2]–[4].

Once a fault occurs, it will propagate to the subprocess or even affect the whole process operating [5]. In principle, the readings of multiple sensors will deviate from the normal status owing to the interaction between different units. Meanwhile, partial variables will respond to different faults, which means one fault category is related to multiple variables and distributed irregularly. These complicated interactions contain the information of nonlinear relationship, correlation, and control rules of the process, which requires a comprehensive modeling technique to explore the fault representation. Therefore, mining interactions and fusing information of multiple sensor measurements are of paramount importance toward fault diagnosis of large-scale industrial processes [6].

The existing methodologies for process industry fault diagnosis can be divided into model-based and data-driven approaches [7]. Due to the complexity of the modern industry, it is difficult to develop the models of the process, which limited the application of the model-based methods [8]. Data-driven methods aim to extract data features and then detect the anomaly variables or recognized the fault categories through statistical analysis [9] or discriminant feature learning [10], [11]. Toward this end, many dimension reduction methods and feature extraction methods are employed such as principal components analysis (PCA) [12], partial least square (PLS) [13], and deep learning (DL) methods including convolutional neural networks (CNNs) [4] and deep belief networks (DBNs) [14]. Although the methods PCA and CCA [15] are classical correlation exploring methods and the CNN methods can extract local relation of data via a kernel function, the interactions between multivariate variables are not explored specifically, which also carry a lot of fault information.

Most of the existing methods take grid data as input ignoring the topology structure of the process and the interaction between monitoring variables, while the data in the graph domain with topology structure are far more common than grid data in practice [16]. As a graph can describe a real-world

Manuscript received May 18, 2021; revised October 1, 2021; accepted November 29, 2021. This work was supported in part by the National Science and Technology Major Project under Grant 2017-I-0007-0008 and in part by the National Natural Scientific Foundation of China under Grant 61732011 and Grant 61925602. *(Corresponding authors: Ruonan Liu; Qinghua Hu.)*

Dongyue Chen, Ruonan Liu, and Qinghua Hu are with the College of Intelligence and Computing and the Tianjin Key Laboratory of Machine Learning, Tianjin University, Tianjin 300350, China (e-mail: dyue_chen@163.com; ruonan.liu@tju.edu.cn; huqinghua@tju.edu.cn).

Steven X. Ding is with the Institute for Automatic Control and Complex Systems (AKS), University of Duisburg-Essen, 47057 Duisburg, Germany (e-mail: steven.ding@uni-due.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3132376.

Digital Object Identifier 10.1109/TNNLS.2021.3132376

system reasonably (e.g. knowledge graph, social network), some nonstructural scenarios are transformed into the graph and use graph-based methods to achieve better performance, including image [17], multiview data [18], and multiagent systems [19]. Due to the complicated interactions between sensor measurements, the structural attributed graph is an appropriate data structure to describe both data characteristics and relationships of sensor data, where each sensor measurement corresponding to a node, and these nodes are linked by an implicit edge representing the interactions. Additionally, the fault information, such as fault root and fault propagation, can be represented in this graph. A critical task of fault diagnosis is to identify the fault categories [10] and thus we formulate this task into a graph classification problem.

While extending fault recognition to the graph classification problem, the following challenge should be settled: 1) a learnable graph construction approach that provides differentiated graph topology for various fault categories and reveals the complex interactions among sensor signals and 2) a model that preserves the specificity of the graph for interaction diversity representation while learns graph representation oriented by fault information. GNNs can be an effective method to fuse the information of multiple sensor readings through a message passing mechanism. In general, exploring the interaction of sensor measurements can be realized by empirical analysis, for example, constructing the graph based on system architecture [16] or the similarity relation in sensor measurements (e.g. K-nearest neighbor graph). Although these graph structures are explainable, they will lead to a similar graph structure between different faults because of the fixed architecture of the system and the fixed dynamical model. A similar graph structure will result in performance degradation of GNN because the features of a node are always aggregated from several fixed neighbors. That means the node features dominate the fault feature learning, where the topology only guide GNN aggregation process and does not provide extra fault information for discriminate learning. To obtain the graph structure varies in fault categories, a supervised graph structure learning approach is employed in this article. Concurrently, in the practical process industry, multiple relations exist between sensor signals, including, but not limited to, similarity, correlation, and functional relationship. For example, in a PS, the feed water pump work and condensate pump work are both calculated by the same equation related to steam mass flow rate, inlet feed water enthalpy, and outlet feed water enthalpy [20]. Consequently, the variables, feed water pump work and condensate pump work, are similar to each other, and furthermore they relate to the other three variables by functional equation. Therefore, one kind of relationship cannot well describe multiple interactions between sensor measurements. In this article, we introduce a heterogeneous graph structure to indicate the complex interactions of sensor measurements, where two nodes are linked by multiple edges.

Based on the above analysis, an intelligent process industry fault diagnosis method is proposed, named interaction-aware graph neural networks (IAGNNs), which considers the multiple interactions between sensor nodes in a graph and fuses the information of each sensor measurements using GNNs. In particular, the attention mechanism is utilized to learn relation scores between nodes in different representation spaces, where a heterogeneous sensor network is constructed. Since the practical relations between nodes are hard to model, we use a relation score to indicate the importance of the interaction between nodes. These learned relation scores constitute the edge attribute of the graph, which provide differences for various fault graphs. The heterogeneous graph can be divided into multiple subgraphs based on edge types. Then, parallel GNN blocks are applied to update the node features for different interaction edge types, respectively, and trained jointly. The updated edge-type-based node embeddings are read out to obtain subgraph representations. These subgraph features are passed through an aggregate function to generate the final graph embedding. Thus, the multivariate time series measured in the process industry can be fused to allow for fault recognition. The major contributions are summarized as follows.

1) This article formulates the fault diagnosis problem of complex industrial processes as a graph classification problem. The key idea is to transform the multivariate sensor signals into a heterogeneous graph with various edge types and classify the fault type taking advance of fused signal embeddings through the message passing mechanism of GNN.
2) The proposed IAGNN framework provides two stages: the graph construction stage and the discriminant feature extraction stage. In the first stage, the attention mechanism is utilized to construct the graph with the consideration of the diversity of edge types and provide differentiate graph topology for various fault categories. In the second stage, the discriminant feature is obtained by fusing the subgraph embeddings from multiple independent GNN blocks. The proposed framework can be used as a generalized platform for industrial process fault diagnosis.
3) Extensive experiments are performed on two industrial processes: a three-phase flow facility and a PS. The experimental results show that the proposed IAGNN framework can provide superior diagnosis results by comparing with the state-of-art methods.

The remaining parts of this article are organized as follows. Section II reviews related works. Preliminaries and problem formulation are elaborated in Section III. The proposed IAGNN method is addressed in Section IV. In Section V, the effectiveness of the proposed method is validated on two complex industrial process datasets. Finally, Section VI concludes this article.

II. Related Work

A. Data-Driven Fault Diagnosis Methods

The data-driven fault diagnosis methods for the industrial processes have been widely studied, which can be broadly classified into two types: 1) multivariate statistical analysis (MSA) methods and 2) learning the discriminate representation of each fault type.

Typical MSA methods for process monitoring and fault diagnosis (PMFD) train a model using data under normal operating conditions, which can convert the high-dimensional

monitoring variables into health indicators and obtain the thresholds to determine the health status of the process. Researchers develop many feature extraction methods according to process characteristics such as linear, nonlinear, and dynamics. Deng *et al.* [12] presented serial principal component analysis for nonlinear process and applied confidence limits to detect fault occurrence. Tao *et al.* [9] proposed dynamic weight PCA in a hierarchical monitoring framework, which gives principal components' different weights according to the separating ability of normal samples and fault samples. Chen *et al.* [21] explored the CCA application for non-Gaussian processes, which derive the threshold by using a randomized algorithm. Most of the existing methods consider the PMFD in a global view, which is difficult to mining the fault root cause for the large-scale process. To overcome this issue, one of the most typical methods is the Bayesian network (BN) [22], [23], which can find the root cause based on stochastic probabilistic theories. Jiang *et al.* [15] proposed a local-global fault monitoring framework integrating deep CCA and distributed modeling technique, which determines the fault occurrence in both local and global views. Li *et al.* [24] proposed a distributed-ensemble model using stacked autoencoders to determine whether the faults affect the operating unit or influence the entire process. Due to the complexity of the modern industry, the fault categories increase explosively. Meanwhile, different faults will lead to similar system responses. Thus, it is hard to meet the requirement of precise fault diagnosis by only using anomaly detection methods.

In order to obtain the discrimination information of different fault categories, FDA, SVM, and CNN are introduced to classify the faults. Sugiyama [25] proposed local FDA (LFDA) for dimension reduction of multimodal data based on the idea of FDA and locality-preserving projection (LPP). Zhong *et al.* [10] extended the LFDA method by exploring the local information of multivariate sensor signals from both sample dimension and variable dimension. Although PCA-based and FDA-based methods have taken into account correlations between variables during the dimension reduction process, these correlations cannot completely reveal the complicated interactions among large-scale sensor measurements. Liu *et al.* [4] developed a multivariate CNN for multivariate time-series classification, in which the lag-feature of time series is considered. Costilla-Reyes *et al.* [26] introduced a CNN-based model to extract novel features from the spatio-temporal tomograph data. However, the complex interactions among sensor signals are not analyzed specifically in these CNN-based methods. Yang *et al.* [6] proposed the Spearman rank correlation-based CNN (SR-CNN) model, which introduces SR correlation to reflect the relationship between sensor signals. In this method, the SR correlation matrices are transformed into images and utilize CNN to extract fault features for classification. Although the SR-image contains the information of relations, the information of raw data is neglected.

B. Graph Neural Networks

The model of graph neural networks (GNNs) aim to extend deep neural networks for structured graph and the concept is first outlined in [27], which fall into the category of recurrent neural network. Recently, the form of GNN to update node embedding is under the framework of message passing neural network [28], by sharing learning across nodes and edges. Kipf and Welling [29] propose graph convolutional networks (GCNs) with message passing operation motivated by a first-order approximation to the spectral graph convolution. Hamilton *et al.* [30] introduce GraphSAGE framework for inductive node embedding by aggregating node features from sampled neighborhood nodes. Another prominent work graph attention network (GAT) [31] introduces the attention mechanisms to message passing operation by learning a shared attention mechanism and compute the importance weights between nodes. Furthermore, the attention mechanisms also extend to heterogeneous graphs. Wang *et al.* [32] proposed that a heterogeneous graph attention network (HAN) consists of node-level attention and semantic-level attention, which introduces the attention mechanism to learn the importance of each meth-path.

Due to the effective application of graph data, GNNs have been introduced to a wide range of scenarios. Fey *et al.* [33] presented a hierarchical message passing framework for molecular property prediction that utilizes two independent GNNs to obtain molecular graph representation on both coarse structure and fine structure. Ying *et al.* [34] proposed a random-walk-based GCN model for the large-scale recommender system. Li *et al.* [35] presented heterogeneous relation attention networks for knowledge graph missing link prediction, which utilizes attention mechanism to learn the importance of each relation path. Meanwhile, the applications of GNNs have also been explored in nonstructural scenarios. Wang *et al.* [36] incorporated a knowledge graph to train a visual classifier for zero-shot recognition. Jia *et al.* [37] presented an attention-based graph learning approach that combines with spatial–temporal GCNs to classify the sleep stage for sleep assessment. Kipf *et al.* [38] presented a neural relational inference (NRI) model to infer an explicit graph structure of an interacting system and learn the corresponding dynamical model. Franceschi *et al.* [39] proposed an approach that optimizes both the parameters of graph generator and GCNs for node classification task. The aforementioned GNN applications can be summarized into three categories that: 1) developed for homogeneous graphs with fixed graph structure principally for node classification task; 2) learn one type edge; and 3) learn multiple edge types for physical system simulation in an unsupervised manner. It can be seen that GNN and its variant methods have been widely applied in many areas, but they still cannot be utilized directly for fault diagnosis of the industrial process because it is formulated as a graph recognition task with the consideration of multiple interactions between nodes.

Recently, GNN-based methods are introduced to mechanical fault diagnosis. Li *et al.* [40] constructed an affinity graph based on the similarity of the vibration signal and proposed a multireceptive GNN for fault diagnosis under node classification task, where a vibration signal segment corresponding to a node. Different from the vibration signal, the monitoring data of the process are multivariate time series. The inner

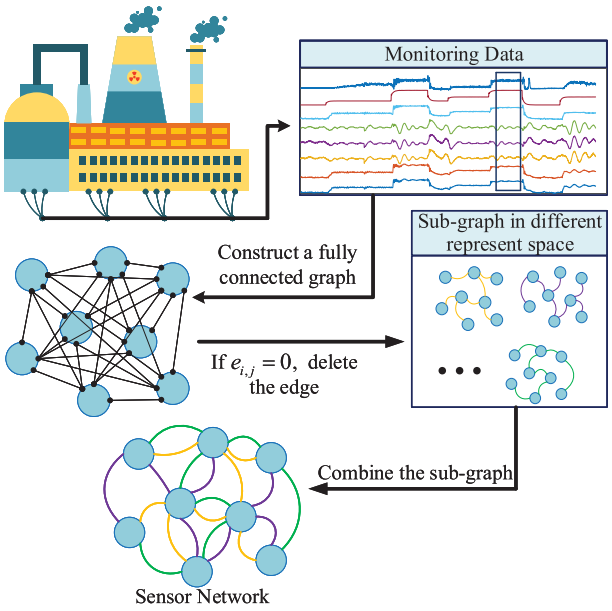


Fig. 1. Graph construction from multisensor readings.

relationship is ignored when we define a segment of the multivariate signals as a node. Wu *et al.* [16] proposed a process topology convolutional network (PTCN), in which the process topology is used to define the graph structure. Chen *et al.* [41] constructed the graph structure based on a model-based system structure analysis method, where the mathematical model is needed. Although both process topology and the system mathematical model can define an interpretable graph with a fixed structure, the topology changes among different fault categories are neglected.

III. Problem Formulation and Preliminaries

In this section, we formally define notations and introduce the message passing operation of GNNs. In this article, the multivariate sensor signals are transformed into graph structure data and Fig. 1 shows an example of the graph construction process. The multivariate sensor signals collected from the industrial process will be cut into segments and form FC graphs with learnable relation scores e for each edge in the graph. After that, the FC graphs are imported into the interaction-aware layer to learn the parameterized relation scores which are trained with the feature extractor jointly. At last, the edge with a zero relation score will be deleted to obtain a sparse graph structure, which can benefit the training efficiency of the GNNs and differential the topology of different fault graphs. In order to better understand the proposed method, we also give the definition of time-series segments and graphs.

A. Problem Formulation

1) Time-Series Segments: The readings from multiple sensors form n raw measured variables, and these time series of length t can be denoted as $\boldsymbol{s}_i = \left(s_i^1, \ldots, s_i^t\right) \in \boldsymbol{S}$. A time series can be cut into segments $\boldsymbol{\omega}_j = \left(\boldsymbol{s}_i^t, \ldots, \boldsymbol{s}_i^{t-m+1}\right) \in \boldsymbol{\Omega}$ through a sliding window with a size of $\boldsymbol{m}$. The window size m is determined according to the stationarity of the time series. Because of the short-term stationarity of sensor signals, the time-series segments are as input to model the graph structure.

2) Graph: A graph is defined as $G = (V, E)$ with vertices $v_i \in V$ and edges $e_{i,j} \in E$, where both nodes and edges can have attributes, noted as $\mathbf{X} \in \mathbb{R}^{n \times d_n}$ and $\mathbf{X}^e \in \mathbb{R}^{c \times d_c}$, respectively. An adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is adopted to record the structure of topology of G. The multivariate time series are converted into graphs under the assumption that there are multiple interaction relationships between these variables. In this article, the sensor network is designed as a heterogeneous graph $G = (V, E, R_E)$, where R_E represents the set of edge types and only one type of node is considered in this article. The attribute of each time-series segment $\boldsymbol{\omega}_j$ is taken as the feature vector $\mathbf{x}_j$ of each node in a graph. It is worth noting that the edge attribute $\mathbf{X}^e$ is obtained through learning.

The research goal is to learn the mapping relationship between the multivariate sensor signals and fault classes based on the proposed method. In this article, we form the fault diagnosis of complex industrial processes as a graph classification problem: given a set of graphs $\{G_1, \ldots, G_N\} \subset \mathcal{G}$ to identify the fault type $\{y_1, \ldots, y_N\} \subset Y$.

B. Basic Graph Neural Network Theory

The GNN extends the deep neural network to process graph structure data. Variants of GNNs have shown powerful performance in node classification tasks, graph classification tasks, and link prediction tasks. In this section, we focus on vertex-centric GNNs that aim to learn a state node representation containing the information of the neighborhood. A GNN takes as input a graph G, where nodes and edges are associated with feature vector $\mathbf{x}_i$ and $\mathbf{x}_{(i,j)}^e$, respectively. The message passing operation in an L-layered GNN has two essential phases, message aggregation, and node embedding update, which can be defined as follows:

$$\mathbf{h}_{(i,j)}^l = \sum_{j \in \mathcal{N}_i} f_{\text{message}}^l \left(h_i^l, h_j^l, x_{(i,j)}^e\right) \tag{1}$$

$$\mathbf{h}_i^{l+1} = f_{\text{update}}^l \left(\mathbf{h}_i^l, \mathbf{h}_{(i,j)}^l\right) \tag{2}$$

where $\mathbf{h}_i$ is the hidden representations of nodes v_i in layer l, $\mathbf{h}_i^{l+1}$ is the updated nodes representations, $\mathbf{h}_{(i,j)}^l$ is the embedding of edge $e_{(i,j)}$, and $\mathcal{N}_i$ is the neighborhood set with an incoming edge to node i. The initial state of $\mathbf{h}_i$ can be set to nodes feature vector $\mathbf{x}_i$. The message function f_{meassage} and update function f_{update} are parametric functions and shared among all nodes, respectively. Each node embedding contains information from its neighboring nodes.

In the node classification task, GNNs classify the label of node v based on the final representation $\mathbf{h}_i^L$. In the graph classification task, GNNs classify the label of a graph based on the global graph representation h_G. Therefore, a readout function f_{readout} is defined as follows:

$$\mathbf{h}_G = f_{\text{readout}}\left(\left\{\mathbf{h}_i^L \mid v \in G\right\}\right). \tag{3}$$

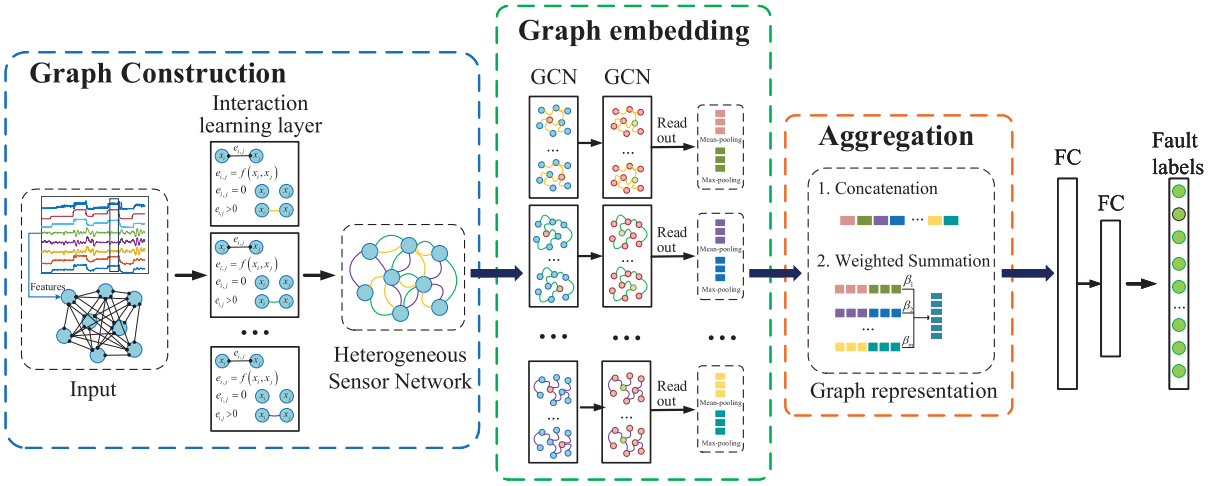


Fig. 2. Framework of the proposed IAGNN.

The readout function f_{readout} can be a simple permutation invariant function, for example, mean function and summation.

IV. Proposed Fault Diagnosis Method

This section proposes an IAGNN model for multivariate time-series classification with application to the industrial process fault diagnosis and its overall framework is illustrated in Fig. 2. The framework is an end-to-end supervised GNN model, which consists of three essential parts and trained jointly: 1) learning pairwise relationship between nodes to construct a heterogeneous graph automatically; 2) leveraging independent GNN for different edge types of the heterogeneous graph to obtain unified nodes representation; and 3) obtain graph representation by reading out the node embeddings and classify various faults.

A. Interaction-Aware Layer

The multiple pairwise interactions between nodes construct an implicit sensor network. According to the domain knowledge, the implicit sensor network can be converted into an explicit graph structure, such as KNN graph and correlation graph based on different correlation metric. Meanwhile, the graph constructed by prior knowledge may not be optimal for model learning due to the subtle difference in node interaction between different fault types. Therefore, in order to differentiate the graph structure of various faults, an attention mechanism-based interaction aware layer is used to learn the explicit graph structure, adaptively. Furthermore, we use multiple independent interaction-aware layers to learn node interactions in different representation spaces, which form the sensor network into a heterogeneous graph with one node type and multiple edge types.

The interaction-aware layer takes sliced multivariate sensor data $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ with an FC graph structure as input and each sensor represents a node. Our goal is to calculate the relation score $r_{i,j}$ to represent the interaction relationship between nodes x_i and x_j. The relation score is used as the edge attribute involved in the message passing process in GNN. In this article, the relational function is defined as a shared single-layer feedforward neural network, parameterized by a weight vector $\overrightarrow{\mathbf{a}} \in \mathbb{R}^{2d}$, which is applied to every node. Then, we perform the attention mechanism to calculate the pth edge relation score $r_{i,j}^p$ between node x_i and x_j with input feature matrix $\mathbf{x} = (x_i^1, \ldots, x_i^m)$. The learned attention coefficients $r_{i,j}$ is defined as

$$r_{i,j}^p = \sigma\left(\overrightarrow{\mathbf{a}}^\top \text{CONCAT}\{\mathbf{x}_i, \mathbf{x}_j\}\right) \tag{4}$$

where $\sigma(\cdot)$ is the active function (e.g. RelU$(\cdot)$) and CONCAT$\{\cdot\}$ operation concatenate $[\mathbf{x}_i, \mathbf{x}_j]$ along feature dimension. The high relation score result in a strong connection between two nodes and the edge between these nodes should be preserved. The relation score with a relatively small value reflects that the interaction of two nodes is weak and the edge between these nodes should be deleted. To make the relation score easily comparable across different nodes and obtain a sparse graph structure, we normalize attention coefficients across all nodes using sparsemax$(\cdot)$ function [42]. The sparsemax transformation returns the Euclidean projection of input onto the probability simplex, which has similar properties to the softmax function, but it can return sparse probability distributions. The normalized relation score $\mathbf{e} \in \mathbb{R}^n$ is defined as

$$\begin{aligned} \mathbf{e} &= \text{sparsemax}(\mathbf{r}) \\ \text{sparsemax}\left(r_{i,j}^p\right) &= \max\left\{0, r_{i,j}^p - \tau\left(r_{i,j}^p\right)\right\} \end{aligned} \tag{5}$$

where $\tau(\cdot)$ is the threshold function that satisfies $\sum_{p=1}^m \max\{0, r_{i,j}^p - \tau(r_{i,j}^p)\} = 1$, and all $r_{i,j}^p$ in relation score set S below the threshold $\tau(\alpha_{i,j})$ will be truncated to zero. Algorithm 1 shows the sparsemax transformation procedure. The sparsemax transformation remains some properties, nonnegative, and $\sum_{p=1}^m \text{sparsemax}(r_{i,j}^p) = 1$ that are common to softmax.

The normalized relation score $\mathbf{e} \in \mathbb{R}$ is used as edge weight of the graph and involves in message passing operations for fault feature extraction. The edge with zero edge weight value is removed and thus a sparse graph structure is obtained.

Algorithm 1: Sparsemax Transformation

Require:
Input relation score vector $\boldsymbol{r} \in \mathbb{R}^n$.
Ensure:
Normalized relation score $\boldsymbol{e}$ with sparse distribution.
1: Sort relation score $\boldsymbol{r}$ as $\boldsymbol{u}$: $u_1 \geq u_2, \ldots, \geq u_n$;
2: Find $q(\boldsymbol{r}) := \max\left\{1 \leq k \leq n \mid 1 + ku_k > \sum_{i \leq k} u_i\right\}$;
3: Define $\tau(\mathbf{z}) = \frac{1}{q(\boldsymbol{r})}\left(\sum_{i \leq q(\boldsymbol{r})} u_i - 1\right)$
4: Compute relation score $\boldsymbol{e} = \max\{0, r_{i,j} - \tau\left(r_{i,j}\right)\}$
5: **return** $\boldsymbol{e}$.

Due to the supervised training process, differentiated graph structures can be obtained. These graph structures guide nodes to aggregate information from different neighbors for different fault categories. In addition, multiple independent interaction-aware layers are employed to reveal the interactions in different perspectives.

B. GCN

GCN has been widely used in various challenging tasks and shows powerful performance. Therefore, GCN is employed as the basic building block to extract fault features. However, it is important to note that other popular GNN architectures can also be employed in our proposed framework, such as GAT [31], GraphSAGE [30], and GIN [43].

The GCN is a graph-based neural network with message passing operations and mathematically is defined as follows:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} c_{ji}\mathbf{h}_j^{(l)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right) \tag{6}$$

where σ is a nonlinearity activation function and $\mathcal{N}(i)$ is the set of neighbors of node x_i. $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ is learnable parameter matrices. $c_{j,i} = 1/(D_{i,i}D_{j,j})^{1/2}$ is normalization constant, and $D_{i,i}$ is the degree of node x_i.

For each set of time-series segments, the interaction-aware layer provides an adjacency matrix $\mathbf{A}$ with an edge weight value for GCN feature extraction. The edge weight $e_{i,j}$ reveals the importance of node x_j to node x_i, especially under different fault classes. Thus, the edge weight should be involved in the message passing process of GCN. If a weight vector on each edge is provided, the normalization constant c_{ji} can be defined as follows:

$$c_{ji} = \frac{e_{ji}}{\sqrt{D_{i,i}D_{j,j}}}. \tag{7}$$

The matrix form of GCN message passing operation is given as follows:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right) \tag{8}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I} \in \mathbb{R}^{N \times N}$ is the adjacency matrix with added self-loops, and $\tilde{\mathbf{D}} = \sum_i \tilde{A}_{ij}$ is the degree matrix of $\tilde{\mathbf{A}}$. The weight value is also considered in the adjacency matrix $\mathbf{A}$. $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times d_l}$ is the activated output of each GCN layer with $\mathbf{H}^{(0)} = \mathbf{x}$, where $\mathbf{x}$ is the feature vector of each node.

The graph G constructed by the interaction learning layer is a heterogeneous graph with multiple edge types. We divide the graph G into K subgraphs according to the number of edge types. For each subgraph, an L-layer GCN block is used to fuse node information, where the parameter of each GCN block need not be shared. These parallel GCN blocks can help keep the independence of the graph structure learning since the interaction-aware layers are trained with these feature extraction blocks, jointly.

C. Read Out

The GCN block updates the node feature of the sensor network, where each node fuses the information of its neighbors. To obtain the global senor network representations, a readout function is utilized to aggregate the node features from the last GCN layer. The readout function considers the global mean properties and local max properties of the sensor network, and thus the readout function is defined as the concatenation of mean-pooling $\mathbf{r}_{\text{mean}}$ and the max-pooling $\mathbf{r}_{\max}$ in kth subgraph

$$\mathbf{r}_k = \text{CONCAT}\{\mathbf{r}_{\text{mean}}, \mathbf{r}_{\max}\}. \tag{9}$$

The mean-pooling and max-pooling readout functions can be described as follows:

$$\mathbf{r}_{\text{mean}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{H}_n^L \tag{10}$$

$$\mathbf{r}_{\max} = \max_{n=1}^{N}\mathbf{H}_n^L \tag{11}$$

where $\mathbf{H}_n^L$ is the node embedding of the last GCN layer, and N is the number of nodes.

In order to preserve the independence of the subgraph representation with different edge types, the nodes embedding only aggregate the information from their neighbors with the same edge type. Therefore, the subgraph representation also indicates a certain perspective in each feature space. One simple and traditional way to obtain a complement graph representation is concatenating vectors across all subgraphs into one vector. Then, we will obtain the graph representation

$$\mathbf{R} = \text{CONCAT}\left\{\mathbf{r}_k \in \mathbb{R}^{2 \times n \times d}\right\}. \tag{12}$$

For subgraph representations from different representation spaces, it is worth noting that the difference between these spaces will not be significant. The redundant information will be introduced to the final graph representation when only use concatenating operation. Meanwhile, when facing a large number of edge types, the concatenated graph representation will lead to the exponential growth of parameters, which can burden the classification of faults. To obtain a more compact graph representation, a weighted summation function is introduced to aggregate the information from subgraphs. First, the embedding of each subgraph is transformed through a nonlinear function, such as MLP, to obtain a transformed embedding. The importance coefficients $\mathbf{w}_{S_k}$ of the transformed embeddings are measured by an attention vector $\overrightarrow{\mathbf{p}}$ and are shown as follows:

$$\mathbf{w}_{S_k} = \sum_{k \in K}\overrightarrow{\mathbf{p}}^{\top} \cdot \text{ReLu}(f_a(\mathbf{r}_k)) \tag{13}$$

Algorithm 2: IAGNN

Require:
The multiple senor signals segments $\mathbf{X} \in \mathbb{R}^{n \times d_n}$;
The Number of edge type to learn K;
Depth of GNNs block layers L.
Interaction learning function Φ_{ia};
GNNs feature extraction block Φ_{gnn};
Value of sub-graph embedding weight β;
Ensure: The graph presentation $\mathbf{R} \in \mathbb{R}^{2 \times n \times d}$.
1: Initialize the graph G with FC structure and initialized the node feature, $\mathbf{H}^{(0)} \leftarrow \mathbf{X}$;
2: **for** $k \in K$ **do**
3: Calculate the relation score, $e_{i,j}^{k} \leftarrow \Phi_{ia}(x_i, x_j)$;
4: Calculate the node embedding, $\mathbf{H}_{k}^{L} \leftarrow \Phi_{gnn}(\mathbf{H}^{(0)}, \mathbf{e})$;
5: Read out the graph embedding of sub-graph with edge type k, $\mathbf{r}^{k} \leftarrow \text{CONCAT}\{\mathbf{r}_{mean}^{k}, \mathbf{r}_{max}^{k}\}$;
6: **end for**
7: Fuse the sub-graph features, $\mathbf{R} \leftarrow \text{CONCAT}\{\mathbf{r}_k\}$, or $\mathbf{R} \leftarrow \sum_{k=1}^{K} \beta_{S_k} \mathbf{r}_k$
8: **return** $\mathbf{R}$.

where f_a is one-layer neural network to obtain the transformed embedding, and $\overrightarrow{\mathbf{p}}$ is the parameterized attention vector. Then the weight β_{S_k} of each subgraph embedding is defined as the normalized importance coefficients using softmax function

$$\beta_{S_k} = \frac{\exp(\mathbf{w}_{S_k})}{\sum_{k \in K} \exp(\mathbf{w}_{S_k})} \tag{14}$$

which represents the contribution of different subgraph embedding for the fault classification tasks. The final graph representation $\mathbf{R}$ can be obtained by fusing the subgraph embeddings with learned weights

$$\mathbf{R} = \sum_{k=1}^{K} \beta_{S_k} \mathbf{r}_k. \tag{15}$$

The number of edge types to learn can be treated as a hyperparameter, and this will be discussed in the experiment.

The fault diagnosis task of industrial processes with multisensor measurement is regarded as a heterogeneous graph classification problem. We apply an FC layer to the final graph representation of the GNN feature extraction module, and the last layer with a softmax classifier outputs the probability of different fault types. We train the parameter with the cross-entropy loss function, denoted as

$$\mathcal{L}(W_f \mid x) = -\sum_{i=1}^{N} p_i \log q_i(W_f \mid x) \tag{16}$$

where W_f is the parameter of the model, p_i is the ground truth, and q_i represents the predicted probability that the instance belongs to fault class i. The overall graph feature extraction process of IAGNNs is shown in Algorithm 2.

The proposed IAGNN method transforms the multivariate sensor signals with implicit structure into explicit graph data and uses the parallel GCN blocks to extract the subgraph features in different representation spaces. The final fault feature fuses multivariate sensor signals and contains the information of complex interactions in a process.

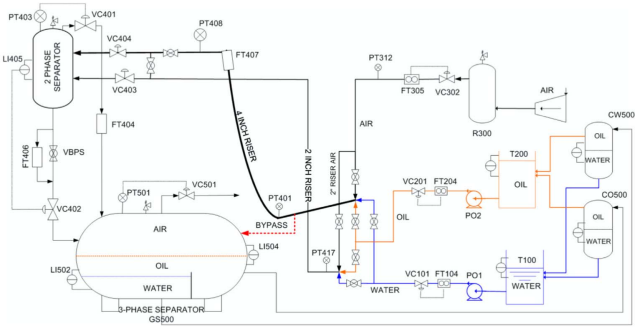


Fig. 3. Architecture of the three-phase flow facility [44].

V. Experimental Study

In this section, we evaluate the performance of the proposed IAGNNs model on two industrial processes data, including a three-phase flow facility simulation dataset and a PS simulation dataset. Extensive experiments and analysis are conducted to demonstrate the following problems: 1) the effectiveness of the proposed method for fault diagnosis of process industry compared with other state-of-art methods; 2) the effectiveness of learned heterogeneous graph compared with fixed graph structure; and 3) the parameter sensitivity analysis of the number of edge types to learn, the depth of GNN blocks, the hidden unit dimension of GNN block, and the embedding dimension of the weighted summation aggregation function.

A. Data Description

1) Three-Phase Flow Facility Data: The three-phase flow facility (TFF) [44] belongs to Cranfield University is designed to control a pressurized system and measure water flow rate, oil flow rate, and airflow rate, which is shown in Fig. 3. There are 24 sensors, measuring pressure, flow rate, density, and temperature at different key locations of the system in Fig. 3, and the detailed information of the sensors can be found in [44]. The dataset is available at https://www.mathworks.com/matlabcentral/fileexchange/50938-a-benchmark-case-for-statistical-process-monitoring-cranfield-multiphase-flow-facility. In order to obtain data under various normal operating conditions, 20 group process inputs were introduced to the simulation and obtain three datasets. For fault datasets, six faults were simulated to indicate typical malfunctions that could occur in real practice. The faults were introduced after a certain time of normal state, when the fault reaching a certain level of severity, the fault state was suspended, and returned to the normal state. Thus, each fault case contains data from weak state to series fault state. Meanwhile, both steady and changing conditions were considered in the simulation process and result in multiple datasets under one fault type. The sampling frequency is 1 Hz. We preprocess each sensor output by max-min normalization, where $\mathbf{x} = (\mathbf{x} - x_{\min})/(x_{\max} - x_{\min})$. We remove the normal data from the fault datasets and take a segment with 50 s of information as one sample. Then we mix all the samples from fault state and normal state and

TABLE I
FAULT TYPES IN THE TFF DATASET

No.	Fault type	Samples of training set	Samples of testing set
1	Air line blockage	127	67
2	Water line blockage	121	51
3	Top separator input blockage	307	126
4	Open direct bypass	153	73
5	Slugging conditions	75	30
6	Pressurization of the 2" line	64	32
7	Normal	478	190

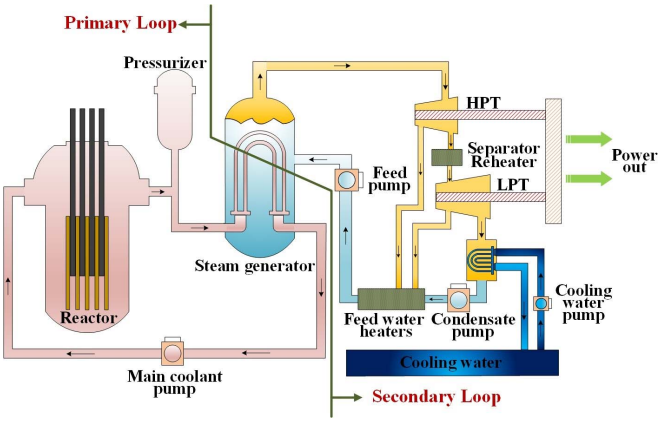


Fig. 4. Simplified schematic of a typical PS.

randomly take 70% samples as training set and 30% samples as testing set. Table I shows the fault types and corresponding sample number of the redivided dataset.

2) Power System Simulation Data: In our experiments, we validate the algorithmic performance on PS data with large-scale fault types and sensor measurements. This PS consists of one reactor and two-loop, primary loop, and secondary loop, with corresponding components, such as a steam generator, a pressurizer, a coolant pump, turbines, a condensate pump, and a feed water pump. The architecture of the PS is illustrated in Fig. 4. The original data have 121 sensor measurements including pressure, flow rate, temperature of main components in the PS, and switch variables of relevant valves. The outputs of the simulator were normalized to [0, 1]. However, consistent with monitoring settings of real practice, the redundant measurements are introduced to the data simulation, which sets more than one sensor at the same location. We clean the dataset by removing the switch variables and the redundant measurements, which results in 76 monitored variables are used in our experiment. In order to obtain normal data covering different operation conditions, six datasets were acquired from the simulator. In the fault simulation process, the faults were set to occur after a certain time of normal state. We choose 53 typical fault cases and normal state data to evaluate the effectiveness of the proposed method. The sampling frequency is 4 Hz. We take a segment with 80 s of information as one sample. Then we mix the data with different categories and divided into training and testing set containing 70% samples and 30% samples, respectively.

TABLE II
FAULT TYPES IN THE PS DATASET

No.	Fault type	Samples of training set	Samples of testing set
1	Fault cases(1-53)	8404	3751
2	Normal	2698	1175

Table II shows the number of samples of the training set and testing set, respectively.

B. Experimental Setup

1) Comparison Methods: To validate the performance of the proposed IAGNN method, we compare our IAGNN method with state-of-the-art baseline methods, including the fault diagnosis method and the graph-based method. The details of the baseline methods are as follows.

1) *PCA+LDA:* PCA+LDA [45]–[47] is a two-stage method, in which the first stage, PCA dimension reduction of raw data, is used to solve the singularity problem of LDA.
2) *SR-CNN:* SR-CNN [6] takes SR correlation images to reflect the variance of the fault feature, and a CNN classifier is applied for fault classification.
3) *PTCN:* PTCN [16] constructs the graph structure via physical connection of the process system and uses GCN to extract the fault feature.
4) *GAT:* GAT learns the edge weights based on the node features. If we convert no structured data into an FC graph, GAT can be applied directly.
5) *PKT-MCNN:* PKT-MCNN [48] introduces a coarse-to-fine framework for large-scale fault diagnosis, in which a hierarchical structure of fault categories is used to guide the knowledge transfer of a multitask CNN model.

2) Implementation Details: To evaluate the performance of the proposed method, the micro F1 score and macro F1 score are adopted in this article. We conduct plenty of experiments for all baseline methods and IAGNN models and choose the hyperparameters which can get the best results. The learning rate of IAGNNs is tuned in the set of $\{0.0001, 0.0005, 0.001\}$, and the number of edge types to learn is sampled from 1 to 8. The IAGNN model for both TFF and PS has two-layer GNN blocks. And, for the TFF dataset, the graph has 24 nodes with an initial feature size of 50, the maximum epoch is 350, the hidden size of GCN and weighted summation module are 128, and three-layer FC with structure $\{256, 128, 7\}$. For the PS dataset, the graph has 76 nodes with an initial feature size of 20, the maximum epoch is 300, the hidden size of GCN and weighted summation module are 512, and three-layer FC with structure $\{1024, 512, 54\}$. The hyper-parameters of the proposed IAGNN model are selected based on the performance and training efficiency and these settings are used to compare the performance with baseline methods. More detailed parameter sensitivity experiments are given in Section V-E. Additionally, the proposed method was implemented using PyTorch geometric on a PC server equipped with NVIDIA RTX 2080Ti and Xeon Silver 4214 CPU.

TABLE III
F1 SCORE (%) COMPARISON BETWEEN GENERATION METHOD AND BASELINE METHODS

Method	TFF		PS	
	Micro F1	Macro F1	Micro F1	Macro F1
PCA+LDA	72.73	70.96	77.78	68.75
SR-CNN	71.70	64.13	72.53	63.63
PTCN	84.71	78.62	77.48	68.41
GAT	85.76	81.63	71.86	61.33
PKT-MCNN	88.58	78.01	86.50	77.29
IAGNN-CON	92.97	89.34	87.47	82.75
IAGNN-AT	92.09	88.07	86.85	81.96

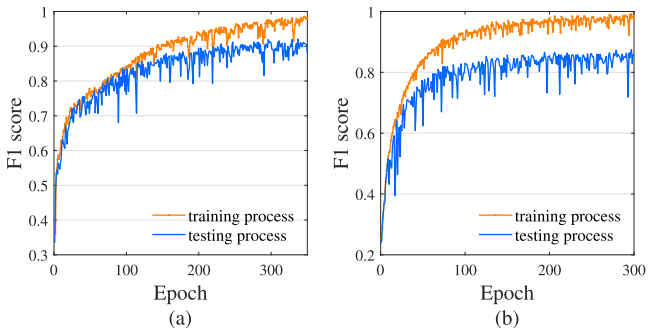


Fig. 5. Convergence of IAGNN on TFF and PS datasets. (a) TFF dataset. (b) PS dataset.

C. Fault Classification Performance

In this article, we follow the practical application that the sample size of normal state data is significantly larger than fault data. Both TFF and PS datasets are imbalanced. The IAGNN model with concatenation operation and weighted summation operation for subgraph feature aggregation are noted as IAGNN-CON and IAGNN-AT, respectively. Fig. 5 shows the overall F1 score curves during the training process and testing process of IAGNN model for both TSS and PS datasets. Although the training curve fluctuates due to sampling imbalance, it can still converge to a stable value and the IAGNN model are not prone to overfitting. The fault diagnosis results of TFF and PS datasets are shown in Table III.

1) TFF Experiment Results Analysis: First, compared with baseline methods, IAGNN models achieve the best performance on TFF dataset. The state-of-the-art results can reveal that the multivariate time-series embedding learned by IAGNNs can effectively reveal the fault characteristic of the process industry. Second, by comparing with the performance of SR-CNN, we can observe that the IAGNN models perform better by taking advantage of both raw data information and interaction information between nodes. This demonstrates that it is necessary to consider the complex interaction between different sensors and embed it in the fault features. Meanwhile, it can be observed that the GAT is outperforming PCA+LDA and SR-CNN methods, which indicates that the GNN can fuse the information of multiple sensor signals more effectively. Third, the IAGNN model obtains better performance than GAT and PTCN, and the reason can be illustrated by two aspects: 1) the FC graph with edge weight without sparse operation will introduce the edge noise to the fault feature, and a fixed graph structure obtained via system physical connection cannot illustrate the differences of faults in the view of topology and 2) the final fault features contain the information from subgraphs in different representation spaces promote the performance of the fault diagnosis.

To further study the performance of each class based on IAGNN models, the fault classification accuracy of various faults for TFF dataset is shown in Table IV. Although the baseline methods have outstanding performance on some fault types, they both fail at some fault categories, such as four baseline methods have poor performance on fault category one. The reduced 2-D feature map of original data and learned

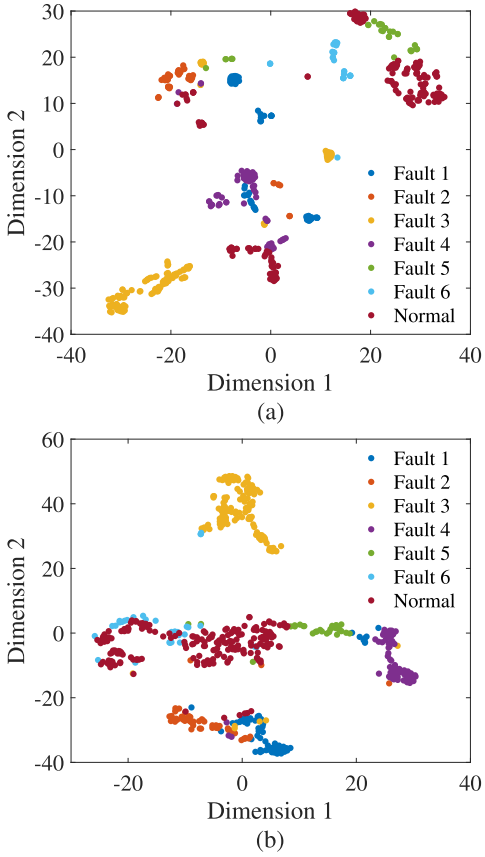


Fig. 6. Visualization via t-SNE. (a) Raw data space and (b) IAGNN-CON learning space.

fault feature in the last layer of the proposed model is shown in Fig. 6 via t-distribution stochastic neighbor embedding (t-sne) method [49]. It can be seen that the sample features of the same fault are separated due to the various working conditions and different fault degrees such as fault 1, while the features of the IAGNN model obtain better cluster performance. In this article, the class imbalance setting is introduced to the datasets. The class with a large number of samples will dominate the cross-entropy loss and the gradient and the class with fewer samples cannot afford model training [50], which is also the reason for the poor fault recognition performance of other

TABLE IV
CLASSIFICATION ACCURACY (%) FOR VARIOUS FAULT TYPES

Fault types	PCA+LDA	SR-CNN	PTCN	GAT	PKT-MCNN	IAGNN-CON	IAGNN-AT
Fault 1	10.48	23.88	76.12	23.88	76.12	88.06	82.09
Fault 2	82.98	49.02	74.51	49.02	84.31	98.04	94.12
Fault 3	68.83	84.13	95.24	84.13	97.62	98.41	96.83
Fault 4	74.99	82.19	87.67	82.19	94.52	87.67	94.52
Fault 5	86.27	83.33	60.00	83.33	96.67	86.67	76.67
Fault 6	88.62	18.75	40.62	18.75	0	65.62	65.62
Normal	88.86	86.84	95.26	99.47	99.47	97.37	97.89

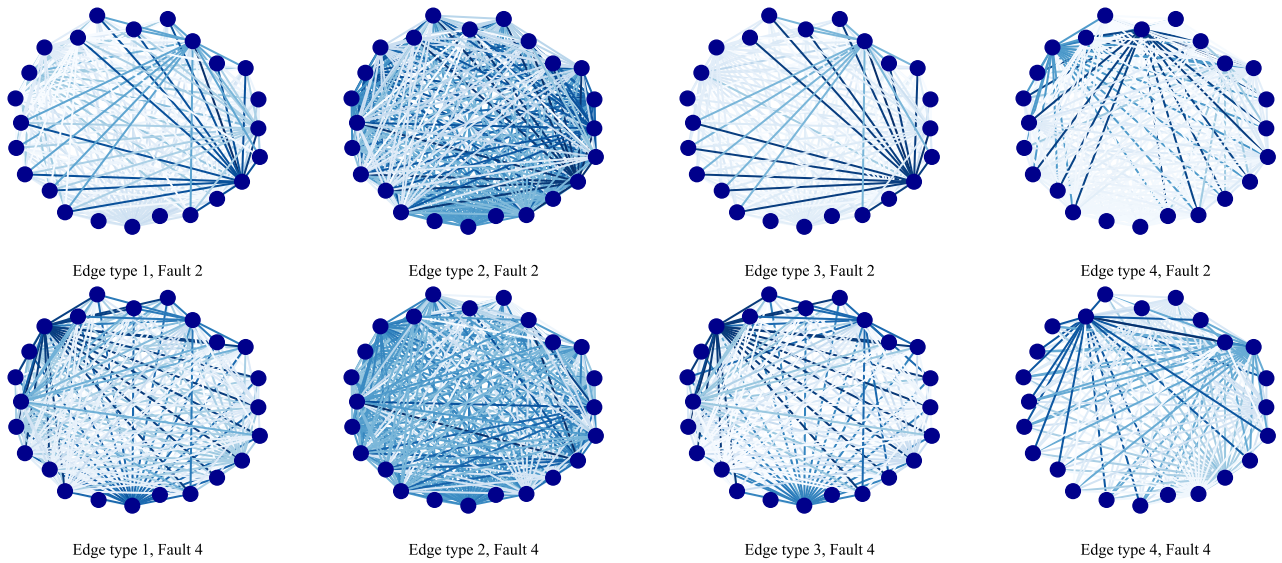


Fig. 7. Graph structure illustration of TFF dataset. The dark blue of the edge indicates a strong connection between nodes.

deep models. However, the proposed IAGNN model has a significant improvement on fault 6. Although PKT-MCNN has the best performance on fault 5, it failed to recognize fault 6 because a well-defined coarse-grained structure is hard to obtain for a small-scale classification task. In general, the performance of IAGNN models is better than the baseline methods, which can obtain acceptable results on all fault categories. The above results prove that the IAGNN models can generate expressive fault representations and the parallel feature blocks can provide comprehensive fault features.

The output of the interaction learning layer is the adjacency matrix with edge weights. Then these adjacency matrices of test samples are added to obtain the learned graph topology. Fig. 7 gives an example of IAGNN models with four learned edge types and shows the graph illustrations of Faults 2 and 4. First, a regular graph structure obtained by the summation of adjacency matrices indicates that the proposed model can learn stable graph structures. Second, the graph output of each interaction-aware layer is different from each other, which indicates that each interaction-aware layer can learn a specific graph structure. Moreover, the learned graphs demonstrate the differences among different fault categories.

2) PS Experiment Results Analysis: In this article, we also apply the IAGNN method to large-scale fault diagnosis of the PS with 54 fault categories and 76 sensor signals. Based on the results in Table III, it can be found that IAGNNs also achieve competitive performance on the PS dataset. The confusion matrices of PS are shown in Fig. 8. It can be observed that the IAGNNs methods have more effective fault diagnosis results than the baseline methods, which indicate that the message passing operation benefits the fusion of multiple sensor signals with multiple interactions. The competitive results of large-scale fault diagnosis tasks further confirm the robustness and effectiveness of the proposed method.

D. Effects of Interaction Learning Layer

In this article, the edge attributes in different representation spaces are gained through the interaction learning layer. To further evaluate the performance of the heterogeneous sensor network learning, a comparison between fixed graph structures is carried out. The different fixed graph construction methods are as follows.

1) FC graph. All the nodes are connected, except the self-connection.
2) K-nearest neighbor (KNN) graph. The adjacency matrix can be obtained by KNN similarity metric, where one node has k neighbor nodes.
3) Maximal information-based nonparametric exploration (MINE) [51] graph. MINE can explore potential

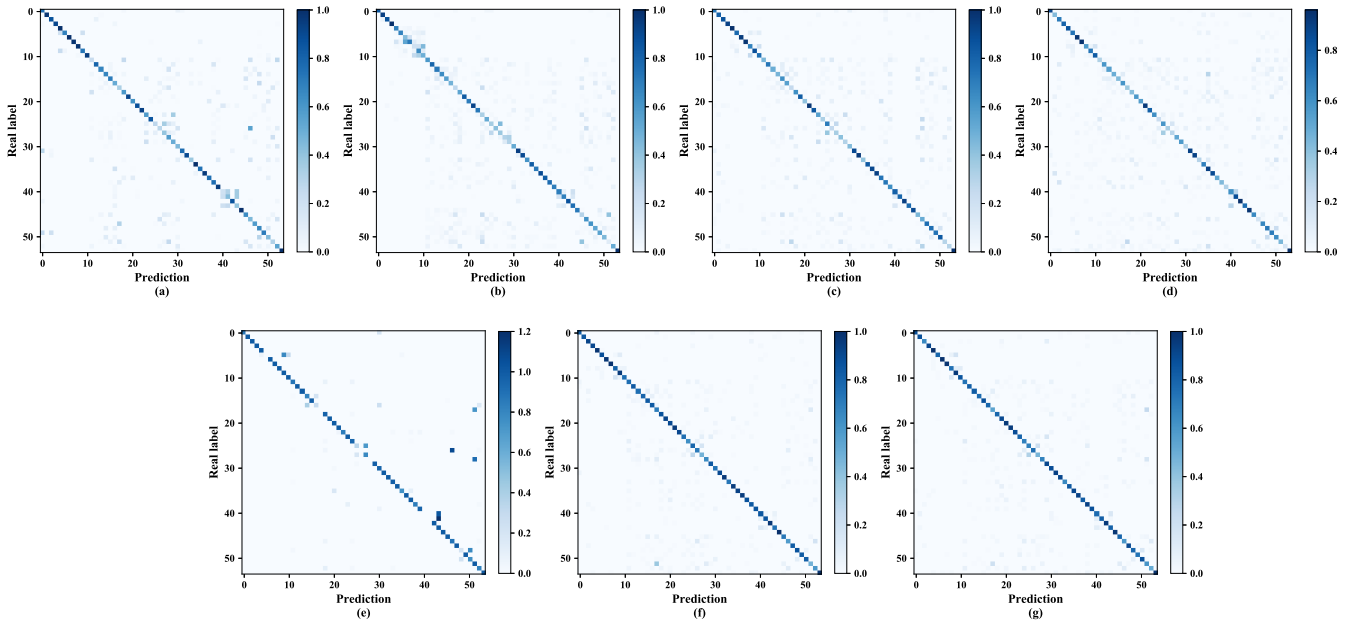


Fig. 8. Confusion matrix of PS. (a) PCA+LDA, (b) SR-CNN, (c) PTCN, (d) GAT, (e) PKT-MCNN, (f) IAGNN-CON, and (g) IAGNN-AT.

TABLE V
F1 SCORE (%) COMPARISON BETWEEN DIFFERENT GRAPH STRUCTURES

Method	TFF		PS	
	Micro F1	Macro F1	Micro F1 score	Macro F1
FC	47.45	24.54	30.59	11.03
MINE	60.11	50.35	81.62	74.24
KNN	87.34	82.87	83.77	77.36
KNN-MINE	90.86	87.79	85.54	79.81
IAGNN-CON	92.97	89.34	87.47	82.75
IAGNN-AT	92.09	88.07	86.85	81.96

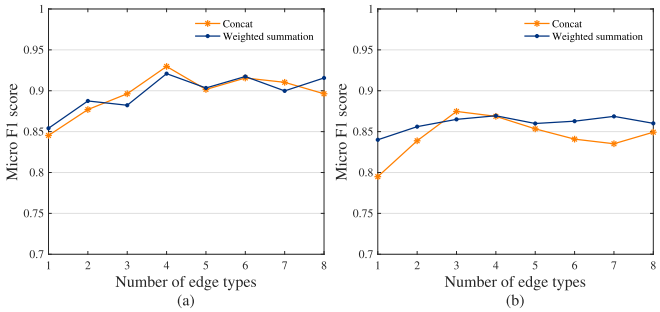


Fig. 9. Influence of different number of edge types on TFF and PS datasets. (a) TFF dataset. (b) PS dataset.

relationships, such as monotonicity and nonlinearity between two variables based on the max information coefficient and the characteristic matrix. It is an efficient tool to identify the structure in data.

4) Hybrid graph. Considering the multirelationship between nodes, KNN and MINE connections are used to construct a heterogeneous graph.

The model structures of these fixed graph experiments are the same as IAGNNs models, except removing the interaction learning layer and subgraph feature aggregation module. The results are shown in Table V. Obviously, the FC graph cannot reveal the relationship between variables, and the embedded fault feature cannot meet the requirement of fault discrimination. Meanwhile, the KNN graph has a better performance than the MINE graph. The results indicate that the KNN structure can better describe the interactions of complex sensor networks than MINE structures. However, the heterogeneous graph with KNN and MINE edge has better results than the graph with one edge type, which demonstrates that the KNN graph and MINE graph can provide complementary information to each other, which can prompt the learning of fault features. Furthermore, IAGNN models achieve the best performance, which proves the superiority of adaptive interaction learning.

E. Parameter Sensitivity

In this part, the impact of: 1) the number of edge types K; 2) the depth of GNN block layers L; 3) the hidden unit dimension of GNN block; and 4) the embedding dimension of the IAGNN-AT aggregation function is investigated.

1) Effects of the Number of Edge Types: Since the IAGNN models learn the interactions between nodes in different representation spaces, the influence of the number of edge types is worth deep analysis. Fig. 9 shows the comparison between the different number of edge types ranging from 1 to 8. First, it can observe that learning multiple interactions receive better performance, which is superior to aggregation of multispace graph information. Second, the best results of IAGNN-CON and IAGNN-AT models are obtained when the number of edges equals 4 for the TFF dataset. After that, the F1 score remained relatively stable with the increase of the number of edges. Moreover, the performance of these two IAGNN models is close to each other. And, this indicates that an

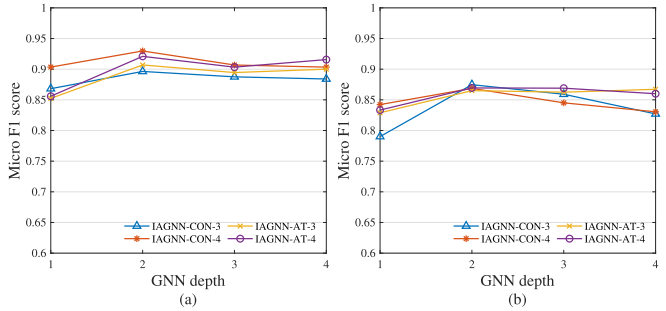


Fig. 10. Parameter sensitivity of depth of GNN layers. (a) TFF dataset. (b) PS dataset.

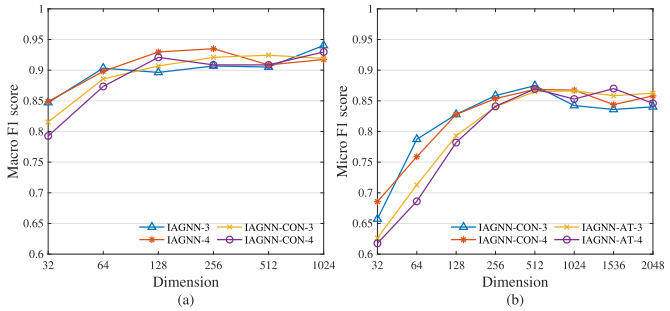


Fig. 11. Parameter sensitivity of hidden unit dimension of GNN layers. (a) TFF dataset. (b) PS dataset.

appropriate choice of the number of edges to learn can improve model performance.

Third, for the PS dataset, the best results of IAGNN-CON are obtained when the number of edges equals to 3 and then the performance decreased. The F1 scores of IAGNN-AT are stable with the increase of the edge types and perform better than IAGNN-CON in most cases. The above results indicate that redundant information is introduced to the final fault features while learning too many edge types and thus burden IAGNN-CON model learning. In contrast, the weight summation aggregation module can capture complementary information from multiple subgraphs, which helps IAGNN-AT obtain better results.

2) Effects of the Depth of GNN Layers: To explore the effect of GNN layers, the experiments with various number of the depth of IAGNN models is carried out and the results of IAGNN-CON models with 3 and 4 edge type and IAGNN-AT models with 3 and 4 edge types are shown in Fig. 10. It can be seen that the best results for the TFF dataset are obtained with the two-layer model. As the depth increases, training a deeper model becomes difficult and the performance becomes worse. The best results for PS data are achieved with the two-layer model, and obvious decrease in the performance of the IAGNN-CON model as the depth of layer increases. Compared to the IAGNN-CON model, the performance of the IAGNN-AT model is relatively stable. This attributes to the weighted summation aggregation function, which provides additional transformation of the subgraph embeddings and contributes to discriminate feature learning.

3) Effects of the Hidden Unit Dimension of GNN: The effect of the hidden unit dimension of GNNs on both the TFF dataset and PS dataset are explored. Fig. 11 shows the results of four IAGNN models, including IAGNN-CON models with three and four edge types and IAGNN-AT models with three and four edge types. In Fig. 11, TFF and PS achieved the expected performance after 128 and 512 hidden unit dimensions, respectively. After that, the performance maintains a certain level. It can be observed that the increase of hidden unit dimensions does help the performance improvement and it needs a wide GNN layer to represent the fault with large-scale categories. Based on the above observation, after obtaining an expected performance, the proposed model is not overly sensitive to the hidden unit dimension.

4) Effects of the Embedding Dimension of IAGNN-AT Aggregation Function: The IAGNN-AT model with a weighted summation function first transforms the embedding of each subgraph through an FC layer. The impact of the size of this FC layer is explored in this part and the results are shown in Fig. 12. The aggregation function with 128 and 512 dimensions achieve the best performance on the TFF dataset and PS dataset, respectively. It can be observed that the embedding dimension of the aggregation function has no serious impact on the fault recognition performance.

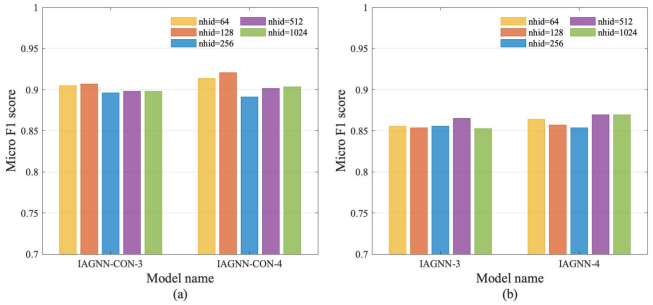


Fig. 12. Parameter sensitivity of embedding dimension of IAGNN-AT aggregation function. (a) TFF dataset. (b) PS dataset.

VI. Conclusion

In this article, we propose an IAGNN model considering the multiple interactions in a sensor network and transform the fault diagnosis problem of complex industrial into a graph classification task. IAGNNs learn the complex interactions between nodes in different representation spaces via an attention mechanism and introduce independent GNN blocks to extract interaction-based subgraph representation and fuse them via an aggregation module to reflect the fault features. Experiences on TFF and PS datasets show that the proposed IAGNNs achieve superior results on each fault category. Furthermore, the effectiveness of adaptively learned graphs and the parameter sensitivity of the number of edge types, the depth of GNN block, the hidden unit dimension, and the embedding dimension of the IAGNN-AT aggregation function are investigated. There are several problems worthy of future work. First, introduce principal variable extraction into message passing operation, such as graph pooling technique, to explore the root cause of the fault and improve the training efficiency of the model. Second, since each node corresponding to a physical measurement variable, the design of node types is worthy for future work, such as distinguishing

the physical variables and the switch variable. Third, learn a graph structure with interpretability, since the edges of the graph learned by the IAGNN model are only described by the relation score.

REFERENCES

[1] Y. Jiang and S. Yin, "Recursive total principle component regression based fault detection and its application to vehicular cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1415–1423, Apr. 2018.

[2] J. Zhu, Z. Ge, and Z. Song, "Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1877–1885, Aug. 2017.

[3] K. Yan, Z. Ji, H. Lu, J. Huang, W. Shen, and Y. Xue, "Fast and accurate classification of time series data using extended ELM: Application in fault diagnosis of air handling units," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 7, pp. 1349–1356, Jul. 2017.

[4] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4788–4797, Jun. 2019.

[5] L. Ma, J. Dong, K. Peng, and C. Zhang, "Hierarchical monitoring and root-cause diagnosis framework for key performance indicator-related multiple faults in process industries," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2091–2100, Apr. 2019.

[6] D. Yang, Y. Pang, B. Zhou, and K. Li, "Fault diagnosis for energy internet using correlation processing-based convolutional neural networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 8, pp. 1739–1748, Aug. 2019.

[7] B. Cai *et al.*, "Remaining useful life re-prediction methodology based on Wiener process: Subsea Christmas tree system as a case study," *Comput. Ind. Eng.*, vol. 151, Jan. 2021, Art. no. 106983.

[8] B. Cai, X. Sun, J. Wang, C. Yang, Z. Wang, and X. Kong, "Fault detection and diagnostic method of diesel engine by combining rule-based algorithm and BNs/BPNNs," *J. Manuf. Syst.*, vol. 57, pp. 148–157, Oct. 2020.

[9] Y. Tao, H. Shi, B. Song, and S. Tan, "A novel dynamic weight principal component analysis method and hierarchical monitoring strategy for process fault detection and diagnosis," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 7994–8004, Sep. 2020.

[10] K. Zhong, M. Han, T. Qiu, and B. Han, "Fault diagnosis of complex processes using sparse kernel local Fisher discriminant analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1581–1591, May 2020.

[11] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *J. Process Control*, vol. 19, no. 10, pp. 1627–1639, 2009.

[12] X. Deng, X. Tian, S. Chen, and C. J. Harris, "Nonlinear process fault diagnosis based on serial principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 560–572, Mar. 2018.

[13] G. Wang and J. Jiao, "A kernel least squares based approach for nonlinear quality-related fault detection," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3195–3204, Apr. 2017.

[14] Y. Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA Trans.*, vol. 96, pp. 457–467, Jan. 2020.

[15] Q. Jiang, S. Yan, H. Cheng, and X. Yan, "Local–global modeling and distributed computing framework for nonlinear plant-wide process monitoring with industrial big data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3355–3365, Aug. 2021.

[16] D. Wu and J. Zhao, "Process topology convolutional network model for chemical process fault diagnosis," *Process Saf. Environ. Protection*, vol. 150, pp. 93–109, Jun. 2021.

[17] V. G. Satorras and J. B. Estrach, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[18] Z. Peng, M. Luo, J. Li, L. Xue, and Q. Zheng, "A deep multi-view framework for anomaly detection on attributed networks," *IEEE Trans. Knowl. Data Eng.*, early access, Aug. 7, 2020, doi: 10.1109/TKDE.2020.3015098.

[19] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Proc. 30th Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4509–4517.

[20] S. M. A. Ibrahim and S. I. Attia, "The influence of the condenser cooling seawater salinity changes on the thermal performance of a nuclear power plant," *Prog. Nucl. Energy*, vol. 79, pp. 115–126, Mar. 2015.

[21] Z. Chen, S. X. Ding, T. Peng, C. Yang, and W. Gui, "Fault detection for non-Gaussian processes using generalized canonical correlation analysis and randomized algorithms," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1559–1567, Feb. 2018.

[22] M. G. Don and F. Khan, "Dynamic process fault detection and diagnosis based on a combined approach of hidden Markov and Bayesian network model," *Chem. Eng. Sci.*, vol. 201, pp. 82–96, Jun. 2019.

[23] B. Cai *et al.*, "Data-driven early fault diagnostic methodology of permanent magnet synchronous motor," *Exp. Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 115000.

[24] Z. Li, L. Tian, Q. Jiang, and X. Yan, "Distributed-ensemble stacked autoencoder model for non-linear process monitoring," *Inf. Sci.*, vol. 542, pp. 302–316, Jan. 2021.

[25] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, no. 1, pp. 1027–1061, Jan. 2007.

[26] O. Costilla-Reyes, P. Scully, and K. B. Ozanyan, "Deep neural networks for learning spatio-temporal features from tomography sensors," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 645–653, Jan. 2018.

[27] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2005, pp. 729–734.

[28] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[30] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1024–1034.

[31] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[32] X. Wang *et al.*, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf. (WWW)*, May 2019, pp. 2022–2032.

[33] M. Fey, J. G. Yuen, and F. Weichert, "Hierarchical inter-message passing for learning on molecular graphs," in *Proc. ICML Graph Represent. Learn. Beyond (GRL+) Workhop*, 2020, pp. 1–6.

[34] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[35] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 19, 2021, doi: 10.1109/TNNLS.2021.3055147.

[36] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6857–6866.

[37] Z. Jia *et al.*, "GraphSleepNet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1324–1330.

[38] T. N. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. S. Zemel, "Neural relational inference for interacting systems," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 2688–2697.

[39] L. Franceschi, M. Niepert, M. Pontil, and X. He, "Learning discrete structures for graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1972–1982.

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

[41] Z. Chen, J. Xu, T. Peng, and C. Yang, "Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge," *IEEE Trans. Cybern.*, early access, Mar. 12, 2021, doi: 10.1109/TCYB.2021.3059002.

[42] G. Li, H. Su, and W. Zhu, "Incorporating external knowledge to answer open-domain visual questions with dynamic memory networks," 2017, *arXiv:1712.00733*.

[43] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks," in *Proc. 7th Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[44] C. Ruiz-Cárcel, Y. Cao, D. Mba, L. Lao, and R. T. Samuel, "Statistical process monitoring of a multiphase flow facility," *Control Eng. Pract.*, vol. 42, pp. 74–88, Sep. 2015.

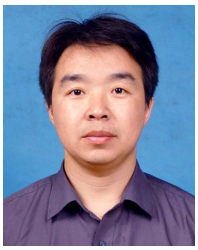
[45] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
[46] J. Ye and Q. Li, "A two-stage linear discriminant analysis via QR-decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 929–941, Jun. 2005.
[47] W. Yu and C. Zhao, "Online fault diagnosis in industrial processes using multimodel exponential discriminant analysis algorithm," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1317–1325, May 2018.
[48] Y. Wang *et al.*, "Coarse-to-fine: Progressive knowledge transfer-based multitask convolutional neural network for intelligent large-scale fault diagnosis," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 9, 2021, doi: 10.1109/TNNLS.2021.3100928.
[49] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.
[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[51] D. N. Reshef *et al.*, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.

Dongyue Chen received the B.S. degree in transportation equipment information engineering and the M.S. degree in precision instruments and machinery from Southwest Jiaotong University, Chengdu, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D. degree in computer applications and techniques with the College of Intelligence and Computing, Tianjin University, Tianjin, China.

Her research interests include deep learning, condition monitoring, and fault diagnosis of mechanical system.

Ruonan Liu (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2013, 2015, and 2019, respectively.

She was a Post-Doctoral Researcher with the School of Computer Science, Carnegie Mellon University in 2019. She currently is an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests include artificial intelligence and machine vision systems.



Qinghua Hu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He was a Post-Doctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently the Dean of the School of Artificial Intelligence, the Vice Chairman of the Tianjin Branch of China Computer Federation, the Vice Director of the SIG Granular Computing and Knowledge Discovery, and the Chinese Association of Artificial Intelligence. He is currently supported by the Key Program, National Natural Science Foundation of China. He has authored or coauthored over 200 peer-reviewed articles. His current research is focused on uncertainty modeling in big data, machine learning with multimodality data, and intelligent unmanned systems.

Dr. Hu is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Acta Automatica Sinica*, and *Energies*.

Steven X. Ding received the Ph.D. degree in electrical engineering from Gerhard–Mercator University, Duisburg, Germany, in 1992.

From 1992 to 1994, he was a Research and Development Engineer with Rheinmetall GmbH, Germany. From 1995 to 2001, he was a Professor of Control Engineering with the University of Applied Science Lausitz, Senftenberg, Germany, and the Vice President from 1998 to 2000. He is currently a Full Professor of Control Engineering and the Head of the Institute for Automatic Control and Complex Systems (AKS) with the University of Duisburg-Essen, Duisburg. His research interests include model-based and data-driven fault diagnosis, fault-tolerant systems, real-time control, and their application in industry with a focus on automotive systems, chemical processes, and renewable energy systems.