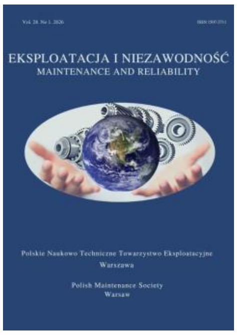


Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 28 (2026), Issue 1

journal homepage: http://www.ein.org.pl

Article citation info:
Zhao Q, Liu X, Li H, Han Y, Multi-Connected Temporal Encoding Graph Neural Network for Complex Industrial Fault Diagnosis, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2026: 28(1) http://doi.org/10.17531/ein/208666

Multi-Connected Temporal Encoding Graph Neural Network for Complex Industrial Fault Diagnosis

Indexed by: Web of Science Group

Qiang Zhao[a,b,*], Xudong Liu[a], Haowei Li[a], Yinghua Han[a,c]

[a] Northeastern University at Qinhuangdao, China
[b] Hebei Key Laboratory of Micro-Nano Precision Optical Sensing and Measurement Technology, China
[c] Hebei Key Laboratory of Marine Perception Network and Data Processing, China.

Highlights

- MCTEGNN: A novel graph neural network for complex industrial fault diagnosis.
- Multi-connected graph captures comprehensive spatial-temporal dependencies.
- Moving-pooling GNN effectively extracts local spatial-temporal features.
- Decay matrix enhances graph accuracy by considering temporal distances.
- Superior performance on TE, TFF and BF dataset validates the model's effectiveness.

Abstract

In modern industry, the complex system scale is large and the process variables are highly coupled. Traditional fault diagnosis methods are inherently challenged in their ability to effectively integrate information, leading to inappropriate feature representations and inaccurate diagnosis outcomes. To address these issues, this paper proposes a Multi-Connected Temporal Encoding Graph Neural Network (MCTEGNN). For multi-connected graph construction, by additionally considering the correlation between different sensors at different timestamps and connecting sensors between all timestamps, a multi-connected graph can be formed to achieve comprehensive dependency modeling. Multi-connected graph convolution captures local dependencies by utilizing moving windows and time pooling, and then learns advanced features to use the moving pool GNN. The proposed method improves the problem of incomplete modeling caused by low efficiency in capturing data relationships. The experimental results on the TE, TFF and BF dataset demonstrate the effectiveness of the proposed MCTEGNN for faults diagnosis.

Keywords
complex industrial processes, fault diagnosis, graph neural network (GNN), Spatial-Temporal relationship model.

1. Introduction

The advancement of equipment integration technology in the industrial sector has led to a notable increase in the maintenance costs associated with operating industrial systems and processes in manufacturing facilities. This is due to the growing number of sensors and instruments that are required for effective monitoring and control [1-3]. Given the complex interaction and intricate connections between the various components, the occurrence of a fault in a specific position may result in the generation of anomalous readings across multiple interconnected components, potentially leading to disruptions in the overall operation of the industrial system. Therefore, it is necessary to accurately and timely identify the type of fault and take corresponding measures. In order to enhance the efficiency, reliability, and safety of the system, research on diagnosing the above-mentioned faults in industry has always been highly concerned.

Considering its physical sequential property together with multiple instrument sensor data, complex industry data exhibits

(*) Corresponding author.
E-mail addresses: Q. Zhao (ORCID: 0000-0003-2004-1769) zhaoqiang@neuq.edu.cn, X. Liu (ORCID: 0009-0001-2141-2552) 2372291@stu.neu.edu.cn, H. Li (ORCID: 0009-0005-6418-5109) 2472307@stu.neu.edu.cn, Y. Han (ORCID: 0000-0003-3768-2212) yhhan@neuq.edu.cn,

a distinctive Spatial-Temporal (ST) dependencies, including temporal correlations between timestamps and spatial correlations between sensors at each timestamp. In addition, the occurrence of different faults can result in the generation of anomalous signals from various components, thereby complicating the process of discerning the relationships between them for industrial operators. Given the above two points, previous manual methods relied on the experience and intuition of operators. This often resulted in failure to meet the requirements for rapid and precise fault diagnosis when faced with highly automated and integrated modern industrial systems. In order to address this situation, it is essential to implement an effective fault diagnosis system. This system would be capable of supervising industrial processes, replacing traditional manual detection methods and reducing costs, thereby protecting industrial infrastructure.

At the present time, the principal methods for diagnosing faults in complex industrial processes are the model-driven and data-driven methods [4-6]. Due to the slow time-varying, distributed parameter, nonlinear, and strong coupling characteristics of complex processes, it is difficult to obtain accurate mathematical models, resulting in suboptimal performance of model driven methods [7]. In the current information age, process data in systems is abundant and easily accessible. Therefore, data-driven methods have become an important direction for dealing with such problems [8].

Numerous studies have demonstrated that the diagnosis of faults in complex industrial processes is fundamentally a classification problem that employs multivariate time-series signals [9]. Therefore, in order to diagnose and analyze faults, it was necessary to rely on the deep architecture of the multi-layer nonlinear data processing unit in the deep learning algorithm for feature extraction [10]. There are complex interactions among components. For example, if a certain part of the system fails, multiple components related to it will generate abnormal signals [11]. The combination of data from multiple components facilitates more effective fault diagnosis. It is therefore essential that, in order to achieve efficient fault diagnosis, the interactions between multiple components are extracted and the hidden information and patterns in time series signals are learned [12]. The majority of traditional data-driven fault diagnosis methods place significant emphasis on the signals of several independent components within industrial systems[13]. However, this approach often fails to consider the interactions between different components, which can lead to limitations in the utilization of extracted features [14].

In order to explore the complex interactions between components, the introduction of graph neural network methods with topological structures can achieve modeling of the interactions between variables [15-16]. Generally, structural attribute diagrams are used to describe industrial process data [17], where each component corresponds to a node, and the edges between nodes represent interaction relationships, which can be learned based on the mathematical characteristics of component signals [18]. The signals emitted by the component in response to different fault conditions are distinct, as are the edges that are learned. Then different topological graph structures are obtained [19-20]. The above is the basic idea of using graph-based methods for fault diagnosis.

Therefore, it is very crucial to develop a graph-based approach that can effectively represent the information contained in component signals and the interactions between components. This can enable the accurate identification of the topology structure of different fault modes. A multitude of methodologies based on graph neural networks have been put forth. Wang et al. [21] employed the short-time Fourier transform on vibration data in order to create a spatial-temporal graph for the purpose of bearing fault diagnosis. Li et al. [22] constructed an affinity graph based on the similarity of the vibration signal and proposed a multi-receptive Graph Neural Network (GNN) for fault diagnosis under node classification tasks. The aforementioned graph-based methods and models have made considerable headway in the domain of fault detection, thereby fostering a plethora of novel insights. However, they still exhibit limitations in their capacity to model spatial-temporal variables. The prevailing approach to graph-based methods is to construct a separate graph for each timestamp, subsequently employing time encoders to capture the temporal relationships between these graphs. The independent processing of information between different time points results in the objective division of complete spatial and temporal relationships. Previous fault diagnosis methods fail to account for the correlation between different sensors at different time points, which constrains their capacity to learn effective representations and real interactions. This results in suboptimal modeling accuracy when confronted with the aforementioned complex coupling problems. The correlations they frequently neglect are pivotal for modeling comprehensive spatial-temporal dependencies within industrial data [23-24], which exhibit intricate physical coupling relationships and temporal correlations [25]. We consider a complex industry fault detection scenario in which a temperature sensor at the inlet of the reactor is highly correlated with a pressure sensor located within the reactor. The time required for material flow means that temperature changes at the inlet of the reactor will not be immediately reflected in the internal pressure [26-27]. In the event of a sudden increase in the temperature of the raw material, the sensor at the inlet is the first to detect this change. However, it takes a certain amount of time for this alteration to affect the pressure within the reactor [28]. In this case, the two sensors are highly correlated at the same timestamp, and the temperature in the initial moments also influences the pressure readings for subsequent moments. This results in correlations between different sensors at different timestamps [29-30]. The constraints of graph construction and graph convolution render existing methods incapable of accurately discerning the interrelationships between disparate sensors at disparate time points [31]. This restricts their capacity to construct comprehensive ST dependencies within intricate multivariate industrial process data [32]. It is necessary to develop a graph structure that can comprehensively consider the relationship between different industrial sensor signals at different time points.

To solve the above limitation, a novel method called Multi-Connected Temporal Encoding Graph Neural Network (MCTEGNN) is proposed, which consists of two key components: multi-connected graph construction and moving-pooling graph convolution. For graph construction, a multi-connected graph is introduced to establish full connections between all sensors across all timestamps. This approach allows us to consider the correlations between sensing variables that do not directly generate a significant relationship. The process commences with the partitioning of each industrial signal sample into discrete patches, with each patch corresponding to a specific timestamp. The sensors situated across all patches are fully connected through dot-product computations. In order to enhance the multi-connected graph, a decay matrix has been devised on the basis of the temporal distances between the aforementioned patches, with greater correlations being attributed to those situated in closer proximity.

Conventional graph-based approaches predominantly focus on two dependency types: intra-timestamp spatial correlations and inter-timestamp temporal continuity. The two relationships are often modeled separately. This fragmented modeling paradigm fails to capture cross-timestamp spatial-temporal interactions (sensor A at t and sensor B at t+Δt), which are critical for industrial processes with material/energy flow delays. Our unified graph architecture overcomes this limitation through full connectivity, enabling holistic modeling of delayed process interactions inherent in complex systems.

Then moving-pooling graph convolution has been devised with the objective of effectively capturing the ST dependencies within the multi-connected graph. A moving-pooling GNN layer is proposed, which employs moving windows of a specified size to slide along patches. In each window, graph convolution is employed to facilitate the updating of node features through edge propagation. Subsequently, a temporal pooling operation is applied to obtain high-level sensor features. Following the application of multiple parallel layers of moving pooling GNN, the updated sensor features are acquired. These are then stacked and mapped in order to obtain the final representations.

The main contributions of our work are summarized as follows:

First, A multi-connected spatial-temporal graph is put forth as a means of explicitly modeling the correlations between sensors across all timestamps. The construction of a temporal distance-based decay matrix facilitates the enhancement of the graph, thereby enabling the effective modeling of the comprehensive spatial-temporal dependencies intrinsic to industrial time series data.

Second, a moving-pooling GNN layer is proposed as a means of effectively capturing the spatial-temporal dependencies inherent to the constructed graph, thereby facilitating the learning of effective representations. The method introduces a moving window for the consideration of local spatial-temporal dependencies, which is then followed by a temporal pooling operation for the extraction of high-level features.

Third, A series of comprehensive experiments have been conducted to demonstrate the effectiveness of our method for effectively modeling and capturing the intricate dependencies inherent to industrial data. Experiments on three datasets demonstrate the effectiveness of MCTEGNN. A comparison with existing fault diagnosis algorithms demonstrates that the MCTEGNN model exhibits enhanced stability in classification and superior generalization ability.

The remaining content of this article is organized as follows: Section 2 elaborates on relevant formulas and preliminary knowledge; Section 3 illustrates the framework of the proposed MCTEGNN fault diagnosis method; In Section 4, the experimental results obtained on complex industrial process datasets were analyzed to verify their effectiveness and reliability; Finally, the conclusion of this article is drawn in Section 5.

2. Formulation and Preliminaries

2.1. GNN for Multivariate Time-Series

In order to more effectively learn the feature representation of multivariate time series data, it is necessary to incorporate spatial dependencies between nodes [33]. To achieve this, a common approach is to leverage GNN, which typically involves the combination of GNN with other temporal encoders [34-35], such as 1DCNN. The general form of a graph convolution operation can be expressed as follows:

$$h^{(l+1)} = \sigma\left(W^{(l)} \cdot \text{MEAN}_{i\in N(j)}\left(h_i^{(l)}\right) + b^{(l)}\right) \tag{1}$$

where $h^{(l)}$ represents the node features at layer l, $N\,(j)$ denotes the neighbors of node j ,and σ is a non-linear activation function. MEANrepresents averaging the features of neighboring nodes. Subsequently, GNNs are employed to capture the spatial dependencies. The spatial dependency can be modeled as follows:

$$H^{(l+1)} = \text{ReLU}\left(W^{(l)} \cdot \text{MEAN}\left(H^{(l)} \cdot A\right) + b^{(l)}\right) \tag{2}$$

Where $H^{(l)}$ are activation value matrix of node features in layer l, A is the adjacency matrix of the graph, and $W^{(l)}$ and $b^{(l)}$ are the learnable parameters. GraphSleepNet [36] also introduced the concept of sequential graphs. In order to extract temporal features, HAGCN [37] employed an LSTM. These features were then used to construct graphs, which were subsequently subjected to further processing by a GNN. These researchers have made a notable impact by employing GNN to identify spatial dependencies within MTS data [38].

2.2. Positional Encoding

Positional encoding is a critical and fundamental component in the Transformer model, which is a type of deep learning architecture commonly used for sequence-to-sequence tasks such as machine translation and language understanding. The Transformer model does not inherently capture the order of the sequence elements. To address this issue, positional encodings are incorporated into the input embeddings, thereby providing the model with information regarding the relative or absolute position of the tokens within the sequence.

The positional encodings are typically generated through the application of sine and cosine functions at different frequencies. The process commences with the generation of a range of positions that correspond to the sequence length. This range is then employed in the computation of the encoding for each dimension of the embedding. For each position, a sinusoidal pattern is created, wherein the frequency of the sine and cosine functions decreases as the dimension index increases. The interleaving of these sine and cosine functions allows for the creation of a comprehensive representation of the positions within the sequence, which can then be learned by the model. Assuming we have a sequence with a length of *len* and each element of the sequence has a dimension of *dmodel*. The position encoding can be represented as a *len* × *dmodel* matrix, where each element is calculated by the following formula:

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{3}$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{10000^{2(i+1)/d_{\text{model}}}}\right) \tag{4}$$

Among them, *pos* refers to the position index within the sequence, and *i* denotes the dimension index. The resulting encodings are subsequently incorporated into the token embeddings, which are then fed into the Transformer’s encoder layers. This incorporation of positional encodings enables the model to comprehend the context and order of the sequence elements.

In practice, the model learns to integrate these positional signals with the semantic information derived from the embeddings in order to make predictions. This method of positional encoding has been demonstrated to be highly effective in enabling the Transformer model to handle long-range dependencies in sequences.

3. METHODOLOGY

3.1. Model Overview

Fig. 1 illustrates the comprehensive framework of the proposed method. The MCTEGNN is designed to integrate the dynamic characteristics of temporal sequences with the spatial structural information, thereby facilitating a precise analysis of industrial processes. Given an industrial data sample, initial processing involves dividing the signal of each sensor into multiple patches, with each patch aligned with its respective timestamp. Each segmented patch is then processed by an encoder module to extract intrinsic features at the sensor level. Subsequently, a position encoding mechanism is adopted with the objective of integrating temporal contextual information into the sensor features of different patches, thereby ensuring that the relative position of sensor data in the time series is taken into account. Next, the construction of multi-connected graph is introduced to achieve comprehensive interconnection between sensors across patches and capture interactions through the calculation of the vector dot product of sensors.

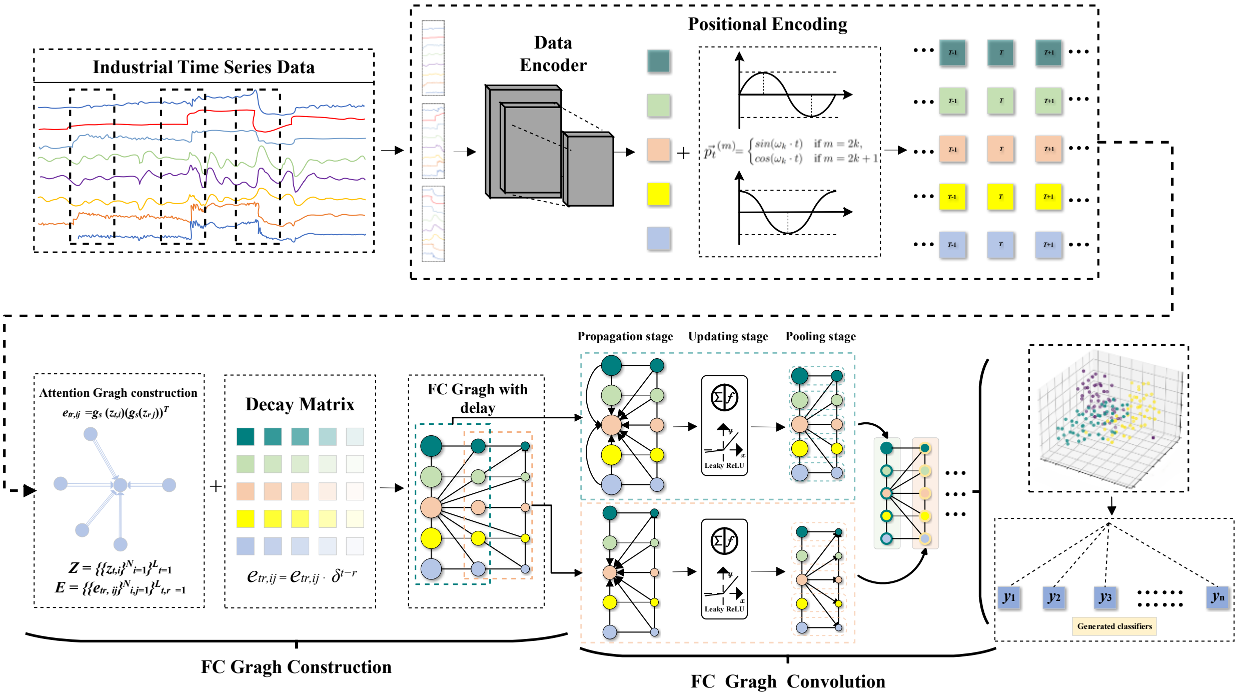

Industrial Time Series Data
Data Encoder
Positional Encoding
$\vec{p_t}^{(m)} = \begin{cases} sin(\omega_k \cdot t) & \text{if } m = 2k, \\ cos(\omega_k \cdot t) & \text{if } m = 2k+1 \end{cases}$
T-1
T
T+1
Attention Gragh construction
$e_{tr,ij} = g_s(z_{t,i})(g_s(z_{r,j}))^T$
$Z = \{\{z_{t,i}\}^N_{i=1}\}^L_{t=1}$
$E = \{\{e_{tr,ij}\}^N_{i,j=1}\}^L_{t,r=1}$
Decay Matrix
$e_{tr,ij} = e_{tr,ij} \cdot \delta^{t-r}$
FC Gragh with delay
Propagation stage
Updating stage
Pooling stage
Leaky ReLU
y_1
y_2
y_3
y_n
Generated classifiers
FC Gragh Construction
FC Gragh Convolution

Figure 1. The overall structure of MCTEGNN.

In order to simulate the correlation between different time nodes in a real system, an attenuation matrix is introduced, taking into account the time distance difference between patches. This reflects the attenuation effect of time on the relationship between different time nodes, thereby more realistically simulating the situation in real industrial processes. Next, a moving-pooling GNN is proposed to capture the ST dependencies from complex industrial systems embedded in the multi-connected graph. We have devised a method of moving windows that traverse along patches. This is then applied to GNN operations within each individual window range, with the aim of capturing fine-grained spatial-temporal dynamics in the data by processing and integrating information segment by segment. After capturing the spatial-temporal dependencies within each window, the sensor features are updated. Subsequently, a temporal pooling operation is applied to extract high-order representations of these sensor features. Finally, the features are input into the classification output layer to obtain the results of subsequent diagnostic tasks. The specifics of the aforementioned methodologies will be elucidated in subsequent chapters.

3.2. Graph Construction

3.2.1. Fully-Connected Graph Construction

Given an industrial sequence data sample $X \in \mathbb{R}^{N\times L}$, the signals of each sensor are segmented into multiple discrete patches. Using a patch denoted by size f, we create$\{X_t\}_{t=1}^{\hat{L}}$ from X, where *t* is the patch index representing a specific timestamp, and each $X_t \in \mathbb{R}^{N\times L}$. L denotes the number of segmented patches, calculated as

$$\hat{L} = \left[\frac{L}{f}\right] \tag{5}$$

where $[\cdot]$ represents the truncation operation to approximate the nearest integer, thereby dividing the time series into uniformly sized discrete intervals.

Each *Xt* contains segmented signals from *n* sensors, i.e.

$$X_t = \{x_{t,i}\}_{i=1}^{N} \tag{6}$$

where $x_{t,i} \in \mathbb{R}^f$. Subsequently, an encoder $f_c(\cdot\,|W_c)$ is employed to process the segmented signals within each window, encoding relevant information for the subsequent feature learning stage. W_c is a set of learnable weight parameters for the encoder. Encoders work at the granularity level of individual sensors to learn intrinsic features of industrial signals, i.e.

$$x'_{t,i} = f_c(x_{t,i}|W_c) \tag{7}$$

In addition, in order to maintain the sequence order between patches, i.e. the inherent relative positional information between patches. Positional encoding is adopted to ensure that the model considers the temporal arrangement of data points.

Specifically, for the *i*-th sensor $\{x'_{t,i}\}_{i=1}^{\hat{L}}$, positional encoding, as shown in Eq. (9), is introduced into sensor features, e.g.

$$z_{t,i} = f_p(t) + x'_{t,i} \tag{8}$$

representing the sensor features enhanced by the incorporation of positional information. By using this processing method, the intrinsic features of sensor information are combined with their relative positions in the time series. Here, *m* denotes the *m*-th feature of sensor features.

$$\overrightarrow{p_t}^{(m)} = f_p(t)^{(m)} := \begin{Bmatrix} sin(wk\cdot t), m = 2k, \\ cos(wk\cdot t), m = 2k+1. \end{Bmatrix} \tag{9}$$

By utilizing sensor features obtained across multiple patches, the correlation between different sensors at different time points is additionally taken into account, and a multi-connected graph is further constructed to interconnect all sensors across these patches, thereby providing a more comprehensive and dynamic representation of complex industrial system behavior. It is assumed that sensors within the same industry that are highly correlated should exhibit similar characteristics, and thus their features should be proximate in the feature space. This allows us to employ the concept of similarity as a means of quantifying the degree of correlation between sensors. It can be posited that the higher the degree of similarity between the features exhibited by the sensors, the stronger the correlation between them.

In this instance, a straightforward and efficacious measurement approach, namely the dot product, is employed to ascertain the degree of similarity between the feature vectors of two sensors, which are defined as follows:

$$e_{tr,ij} = g_s(z_{t,i})(g_s(z_{r,j}))^T \tag{10}$$

where *t, r* ∈ [1, *L*^] and *i, j* ∈ [1, *N*]. The dot product avoids the computational overhead of repeated L2-norm calculations required for cosine similarity. Moreover, the gradient of dot product can help avoid the gradient explosion/disappearance caused by denominator in cosine similarity. Here, the function $g_s(z) = zW_s$ is used to enhance the expressive power of the model, where W_s is a learnable weight matrix. In addition, the softmax function constrains the value of correlation to a closed interval of [0,1].

Finally, the multi-connected graph $\mathcal{G} = (Z, E)$ is derived, where

$$Z = \left\{\{z_{t,i}\}_{i=1}^{N}\right\}_{t=1}^{\hat{L}} \tag{11}$$

and

$$E = \left\{\{e_{tr,ij}\}_{i,j=1}^{N}\right\}_{t,r=1}^{\hat{L}} \tag{12}$$

E represents the adjacency matrix of the multi-connected graph, which quantifies the correlation between sensors across all patches and comprehensively models the inherent dependencies of industrial time series. The graph *G* not only includes the temporal correlation of sequential dependencies between different timestamps and the spatial correlation between sensors within the same timestamp, but also the correlation between different sensors at different timestamps. The combination of these three relationships enables us to construct a comprehensive model of the ST correlation in multivariate industrial time series data in detail, which is crucial for accurately interpreting the inherent complex interactions and patterns in multivariate industrial time series data, thereby enhancing the predictive ability and analytical depth of graph-based methods.

3.2.2. Decay Matrix

The multi-connected graph G is constructed based on sensor similarity across patches only, without accounting for temporal distances between sensors across these patches. Motivated by this, a decay matrix has been developed which incorporates temporal distances between sensors, with the objective of enhancing the precision of the graph G . The dimension of this adjacent matrix is$E \in \mathbb{R}^{(3\times4)\times(3\times4)}$. In this matrix, each row represents the connections between sensors across all patches. The initial four columns illustrate the interconnectivity between sensors within the same patch. Due to the fact that these data come from industrial instrument data with the same timestamp and are influenced by the same physical processes or conditions occurring at specific time points, they should exhibit stronger correlation than data in patches with different timestamps and should be represented with higher correlation values in the adjacency matrix.

As the values of these sensors are located at different relative time positions, their correlation with $z_{T-1,1}$ should decay, and the time difference should be quantified by the decay rate. As the temporal gap widens, the correlation estimation between sensors will naturally further decrease, measured by δ^2 to accurately simulate the weakening of the influence of farther industrial parameters on each other over time. We formulate the decay matrix

$$C = \left\{\{c_{tr,ij}\}_{i,j=1}^{N}\right\}_{t,r=1}^{\hat{L}} \tag{13}$$

where each element $c_{tr,ij} = \delta^{t-r}$. This matrix is employed to refine and enhance the correlations between sensors across patches, yielding $e_{tr,ij} = e_{tr,ij}c_{tr,ij}$ corrected to scale the initial correlation.

3.2.3. Moving-pooling Graph Convolution

The subsequent stage is to employ the constructed multi-connected graph in order to capture the ST dependencies within complex industrial data for representation learning. A methodology that has been frequently employed in the processing of data sets is the application of graph convolution across the entire constructed graph. This approach may prove inadequate for effectively capturing the local ST dependencies within industrial process data with a large number of transient processes.

In order to address the limitations, a moving pooling GNN is proposed, which incorporates a moving window to capture local ST dependencies and temporal pooling to extract high-level features. Unlike standard pooling approaches that apply global aggregation over the entire sequence, the moving-pooling operates within local temporal windows. For each window spanning M timestamps, we compute the average pooling only within the current window (e.g., $t \in [w-\frac{M}{2}, w+\frac{M}{2}]$ for window w). This preserves local temporal patterns critical for industrial fault diagnosis while avoiding over-smoothing from global aggregation.

Specifically, following previous works, a Message Passing Neural Network (MPNN), a variant of GNN, is utilized to capture the ST dependencies of the graph within each window. The MPNN comprises both a propagation and an update stage. During the propagation stage, the information from neighboring nodes is propagated into the central node. Given a central node $z_{t,i}^{l}$ of the w-th window in the l-th layer, it has a set of neighboring nodes $\left\{\{z_{r,j}^{l}\}_{j=1}^{N}\right\}_{r=w-\frac{m}{2}}^{w+\frac{m}{2}}$across M patches in the same window, w represents the index of the window currently being processed.

The central node has correlations with its neighbors as $\left\{\{e_{tr,ij}^{l}\}_{j=1}^{N}\right\}_{r=w-\frac{m}{2}}^{w+\frac{m}{2}}$. After the propagation stage, we obtain the propagated features

$$h_{t,i}^{l} = \sum_{r=w-\frac{m}{2}}^{w+\frac{m}{2}} \sum_{j=1}^{N} z_{r,j}^{l}\, e_{tr,ij}^{l} \tag{14}$$

Then, the updating stage employs a non-linear function to update the propagated sensor features, i.e.,$z_{t,i}^{l+1} = f_g(h_{t,i}^{l}|W_g)$.

Given the updated sensor features $\{z_{t,i}^{l+1}\}_{t=w-\frac{M}{2}}^{w+\frac{M}{2}}$ for the i-th sensor across M patches, temporal pooling is conducted using an average pooling strategy, yielding sensor features

$$z_{w,i}^{l+1} = \sum_{t=w-\frac{M}{2}}^{w+\frac{M}{2}} z_{t,i}^{l+1}/M \tag{15}$$

for the *w*-th window. Subsequently, the sensors are stacked across all windows, thereby creating a high-level multi-connected graph that serves as the input for the subsequent layer. It should be noted that this study employs a single layer, whereby the sensor features obtained from each slide window are directly utilized for output purposes.

The direct output of MCTEGNN is a representation vector for each input industrial time-series sample. For the subsequent fault diagnosis task, this representation serves as input to a softmax classifier, producing a probability distribution over fault classes.

4. Results and Discussion

In this section, the effectiveness of the proposed model MCTEGNN will be tested and comprehensively evaluated on three industrial datasets, and compared with other existing baseline models for fault diagnosis.

4.1. Datasets Description

1)Tennessee Eastman (TE) Process: The Tennessee Eastman Process, developed by Eastman Chemical Company, has gained considerable recognition as a popular simulation platform for the detection and diagnosis of faults.

This dataset simulates actual chemical processes for anomaly detection and process adjustment. The entire process is comprised of five distinct operational units, including the reactor, condenser, gas-liquid separator, circulating compressor, and product stripping tower. Table 1 and Table 2 provide detailed information on each variable and fault type. The process flowchart is presented in Figure 2.

This article employs a revised simulation model to generate experimental data, and further detailed information is available at [40]. Only the original 41 measurement values and 9 manipulated variables are collected, and variables (5, 9, and 12) were excluded because they remained unchanged during the simulation process and therefore did not participate in the fault detection diagnosis calculation. For a normal state, the process was simulated for a period of 500 hours. In order to obtain a representative sample, the process was simulated 10 times for each fault state, with different random initial values. Each simulation lasted 100 hours. In each instance of the fault state simulation, the disturbance was introduced following the completion of eight hours of normal process operation. In particular, fault 6 will result in the process industry system being terminated after the introduction of interference for a period of seven hours. Consequently, each simulation of fault 6 is limited to seven hours of fault data. The sampling interval has been established at three minutes. A single sample is constituted by a segment containing one hour of information. Subsequently, all samples were combined with the fault state and normal state, and a random selection was made of 80% for the training set and 20% for the testing set.

Table 1. Monitoring variables in TE process.

ID	Description	ID	Description
XMEAS1	A Feed (Flow 1)	XMEAS28	Component F (Flow 6)
XMEAS2	D Feed (Flow 2)	XMEAS29	Component A (Flow 9)
XMEAS3	E Feed (Flow 3)	XMEAS30	Component B (Flow 9)
XMEAS4	Total Feed (Flow 4)	XMEAS31	Component C (Flow 9)
XMEAS5	Recycle Flow Rate (Flow 8)	XMEAS32	Component D (Flow 9)
XMEAS6	Reactor Feed Rate (Flow 6)	XMEAS33	Component E (Flow 9)
XMEAS7	Reactor Pressure	XMEAS34	Component F (Flow 9)
XMEAS8	Reactor Level	XMEAS35	Component G (Flow 9)
XMEAS9	Reactor Temperature	XMEAS36	Component H (Flow 9)
XMEAS10	Emission Rate (Flow 9)	XMEAS37	Component D (Flow 9)
XMEAS11	Product Separator Temperature	XMEAS38	Component E (Flow 9)
XMEAS12	Product Separator Liquid Level	XMEAS39	Component F (Flow 9)
XMEAS13	Product Separator Pressure	XMEAS40	Component G (Flow 9)
XMEAS14	Product Separator Outlet Flow Rate (Flow 10)	XMEAS41	Component H (Flow 9)
XMEAS15	Stripping Tower Liquid Level	XMV1	D Feed Flow Rate (Flow 2)
XMEAS16	Stripping Tower Pressure	XMV2	E Feed Flow Rate (Flow 3)
XMEAS17	Stripping Tower Bottom Flow (Flow 11)	XMV3	A Feed Flow Rate (Flow 1)
XMEAS18	Stripping Tower Temperature	XMV4	A and C Feed Total Flow Rate (Flow 2)
XMEAS19	Stripping Tower Flow Rate	XMV5	Compressor Recycle Valve

ID	Description	ID	Description
XMEAS20	Compressor Working Power	XMV6	Emission Valve (Flow 9)
XMEAS21	Reactor Cooling Water Outlet Temperature	XMV7	Liquid Filling Flow Rate (Flow 10)
XMEAS22	Separator Cooling Water Outlet Temperature	XMV8	Tower Liquid Product Flow Rate (Flow 11)
XMEAS23	Component A (Flow 6)	XMV9	Stripping Tower Water Flow Valve
XMEAS24	Component B (Flow 6)	XMV10	Reactor Cooling Water Flow Rate
XMEAS25	Component C (Flow 6)	XMV11	Condenser Cooling Water Flow Rate
XMEAS26	Component D (Flow 6)	XMV12	Agitation Rate
XMEAS27	Component E (Flow 6)		

Table 2. Fault descriptions in the TE process.

ID	Fault Description	Type
IDV1	A/C feed flow ratio changes, B content remains unchanged (Stream 4)	Step
IDV2	B content changes, A/C feed flow ratio remains unchanged (Stream 4)	Step
IDV3	Material D temperature changes	Step
IDV4	Reactor cooling water inlet temperature changes	Step
IDV5	Condenser cooling water inlet temperature changes	Step
IDV6	Material A loss (Stream 1)	Step
IDV7	Material C pressure loss (Stream 4)	Random Variable
IDV8	A, B, C composition changes (Stream 4)	Random Variable
IDV9	Material D temperature changes (Stream 2)	Random Variable
IDV10	Material C temperature changes (Stream 2)	Random Variable
IDV11	Reactor cooling water inlet temperature changes	Random Variable
IDV12	Condenser cooling water inlet temperature changes	Slow Drift
IDV13	Reaction kinetics characteristics change	Sticky
IDV14	Reactor cooling water valve stick	Sticky
IDV15	Condenser cooling water valve stick	Sticky
IDV16	Unknown	Unknown
IDV17	Unknown	Unknown
IDV18	Unknown	Unknown
IDV19	Unknown	Unknown
IDV20	Unknown	Unknown

Table 3 illustrates the sample sizes of the training and testing sets employed for the observed data presented in this article.

Table 3. Fault type in the TE dataset.

NO.	Fault type	Training samples	Testing samples
1	Fault 1-5&7-20	304000	76000
2	Fault 6	1120	280
3	Normal	16000	4000

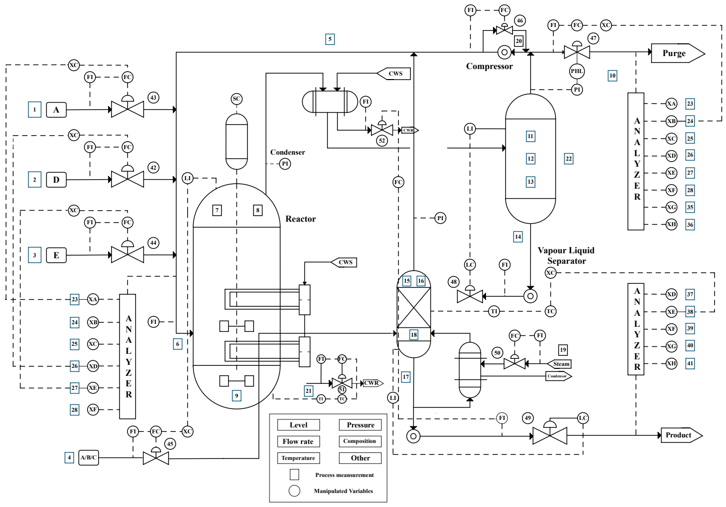


Figure 2. Process flow of the TE process.

2) Three-Phase Flow Facility (TFF) Process: The Three-Phase Flow Facility Process [41] designed by Cranfield University, is a realistic and complex industrial process platform in which the primary function is to obtain experimental data for fault diagnosis by providing controlled and measured flows of water, oil, and air to a pressurized system. This dataset contains normal states and six different fault states. We combine all normal and faulty subsamples and randomly take 70% as the training set and 30% as the testing set. The redivided dataset is shown in Table 4, which includes the fault types and the corresponding number of samples.

Table 4. Fault type in the TFF dataset.

NO.	Fault type	Training samples	Testing samples
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

3) Blast Furnace (BF) Process: The modern large-scale blast furnace smelting process includes multiple key subsystems such as feeding system, hot blast furnace system, coal injection system, and blast furnace body. The blast furnace body consists of five parts: throat, body, belly, waist, and hearth. It is the core link of the ironmaking process, and its working principle is based on the top-down movement of furnace materials and the dynamic process of reverse continuous reduction of iron ore by gas flow from bottom to top. Blast furnace smelting is a complex dynamic process that involves multiple systems and reactions. The invisibility of its internal state, the complexity of its operational steps, and the intersection of physical and chemical processes make direct observation and evaluation of it a major challenge. In order to further verify the applicability of the proposed method in actual industrial environments, this study studied and explained the five sampled fault states based on actual process data collected and calculated from a large steel enterprise's blast furnace smelting site. The industrial parameters involved are shown in Table 5. The fault description and sample partitioning of this dataset are shown in Table 6. The proportion of the training set in this dataset is 70%, and the proportion of the testing set is 30%.

Table 5. Variables in BF process.

ID	Description	ID	Description
1	Furnace coke_Ad	17	Spray rate
2	Furnace coke_Mt	18	Oxygen enrichment level
3	Furnace coke_St	19	Furnace top pressure
4	Injection of coal_Ad	20	Coke load
5	Injection of coal_Fcad	21	Mean value of probe
6	Injection of coal_St	22	Range of measuring tape
7	abrasion resistance index	23	Material speed
8	drum index	24	Standardized wind speed
9	Strength of Charcoal Reaction_CSR	25	Hot air pressure
10	Coke Reactivity_CRI	26	Mean throat temperature
11	Coke compressive strength: M40	27	Temperature range of furnace throat
12	Coke anti-wear index _ M10	28	Mean temperature of furnace top
13	Coke batch weight	29	Range of furnace top temperature
14	Ore batch weight	30	Gas utilization rate
15	Hot air temperature	31	Total area of air outlet
16	Supply air volume		

Table 6. Fault type in the BF dataset.

NO.	Fault type	Training samples	Testing samples
1	Fluctuations in supply air volume	830	357
2	Abnormal mass fraction of fixed carbon in coal powder injection	571	246
3	Abnormal supply of oxygen enrichment	2184	936
4	Abnormal metallurgical coke	1092	468
5	Furnace throat temperature fluctuation	1528	656

In this article, the micro F1 score and macro F1 score are employed to validate the effectiveness of the proposed MCTEGNN and integrate five baseline methods for comparison, namely GAT [42], TAGCN [43], MRFGCN [44],IAGNN [44] and MTGNN [45]. Throughout the entirety of the training process, the optimizer is Adam, with an initial learning rate of 0.0001. All networks, including the baseline methods, were implemented using Pytorch 2.0.1 and Python 3.8.2 on Ubuntu 18.04, equipped with NVIDIA GeForce RTX 3090Ti GPU 24GB of RAM.

4.2. Classification Result Analysis of TE Experiment

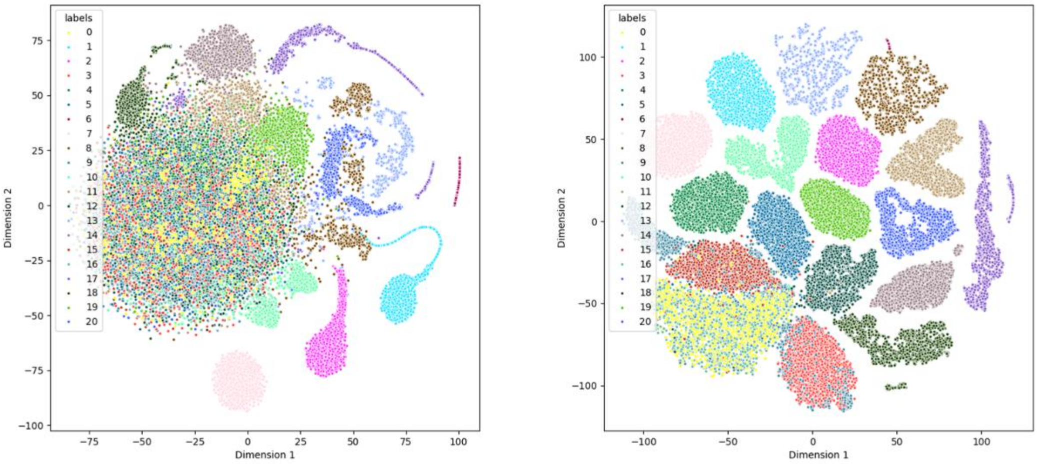
The detailed evaluation results of the complex industrial process time series dataset are shown in Table 7. We perform fault diagnosis classification on the TE dataset based on six graph neural deep learning models that can be used for industrial fault diagnosis.

Comparing with the results of several baseline models including GAT, TAGCN, MRF-GCN, IAGNN and MTGNN, it can be observed that MCTEGNN achieved the best or comparable performance. From the classification results of 20 faults and normal states that need to be diagnosed in the TE dataset, it can be seen that the proposed MCTEGNN achieves the best diagnostic accuracy among the six diagnostic methods in 16 fault categories. Compared with the baseline methods, the proposed MCTEGNN achieved the best classification results, with the highest micro F1 score of 0.9527, and the macro F1 score is 0.9554. This provides further evidence of the effectiveness of the model proposed in this article. Figure 3 illustrates the feature space distribution of the model data mapped to a low dimensional space before and after classification.

Table 7. Classification Accuracy (%) for Various Fault Types on The TE Dataset.

Fault types	GAT	TAGCN	IAGNN	MRF-GCN	MTGNN	**MCTEGNN**
Normal	82.8	91.18	89.95	84.59	82.38	89.54
Fault 1	100	100	100	100	100	**100**
Fault 2	98.12	99.94	100	100	100	**100**
Fault 3	45.98	94.56	88.63	89.18	77.14	**98.68**
Fault 4	99.85	99.62	99.61	100	100	**100**
Fault 5	52.1	80.66	65.24	66.09	60.45	**87.31**
Fault 6	100	100	100	100	100	**100**
Fault 7	99.85	100	100	100	100	**100**
Fault 8	73.43	99.79	99.54	99.35	94.98	**99.81**
Fault 9	46.2	66.84	51.32	38.72	39.63	**78.68**
Fault 10	74.45	97.19	97.51	97.06	94.02	**99.15**
Fault 11	93.48	99.72	99.15	99.68	79.79	**100**
Fault 12	81.75	98.61	94.67	95.00	88.07	**98.85**
Fault 13	77.92	96.85	96.18	96.93	87.52	**97.89**
Fault 14	99.85	100	100	100	90.52	**100**
Fault 15	48.12	72.04	61.76	56.81	50.79	**88.53**
Fault 16	28.57	58.73	52.69	31.98	76.95	76.54
Fault 17	93.1	96.55	94.49	97.11	89.27	**97.17**

Fault types	GAT	TAGCN	IAGNN	MRF-GCN	MTGNN	**MCTEGNN**
Fault 18	67.85	96.01	95.63	94.64	88.84	95.85
Fault 19	89.6	99.42	99.53	99.26	89.61	**100**
Fault 20	91.97	97.34	96.02	97.30	95.59	97.19



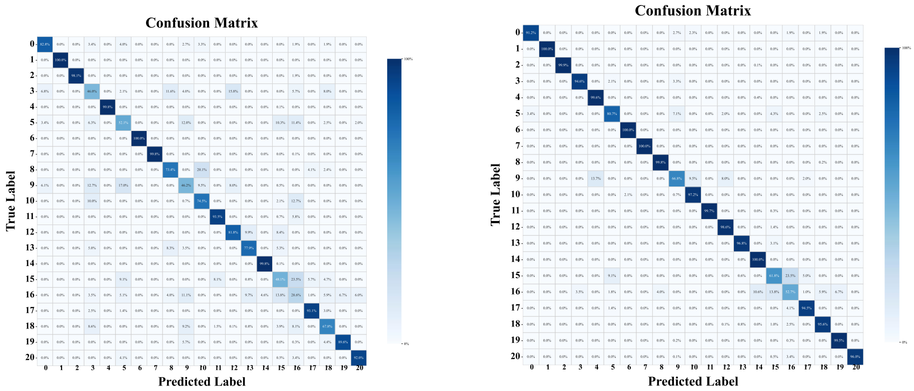
(a)

(b)

Figure 3. The visualization distribution of data features after classification by the model is shown using t-SNE graphs [39], where Figure (a) represents before classification and Figure (b) represents after classification.

The significant improvement achieved in the results is that even for the most intricate faults (3, 5, 9, 15, and 16), the fault diagnosis rate of MCTEGNN has significantly improved compared to previous methods. The identification of these five types of faults has always been a challenging problem, with relatively low diagnostic accuracy achieved in existing models. Overall, it can be seen from Table 7 that the results obtained by the proposed model are comparatively superior to those of the baselines. Through analysis, it can be concluded that the reason for the improvement is that the proposed model adopts an algorithm modeling that is more in line with the spatial-temporal relationships between complex industrial process variables. The design of multi-connected graphs enables the model to comprehensively capture spatial-temporal dependencies in complex industrial data, thereby enhancing its capacity for generalization across different fault modes. Mobile pooling graph convolution also enables models to more effectively comprehend and process local patterns and dynamic changes in industrial time series data. This experiment serves to further demonstrate the powerful capabilities of MCTEGNN in data analysis and classification.

Diagnostic accuracy for Faults 9 and 15 remains suboptimal compared to other faults due to their unique physical characteristics: Fault 9 manifests as gradual parameter shifts that are masked within normal process variations. The model's 78.68% accuracy reflects challenges in distinguishing subtle, long-term trends from noise. Fault 15 exhibits complex nonlinear behavior, causing irregular parameter fluctuations. The 88.53% accuracy indicates residual difficulties in modeling these state transitions.



a)

(b)

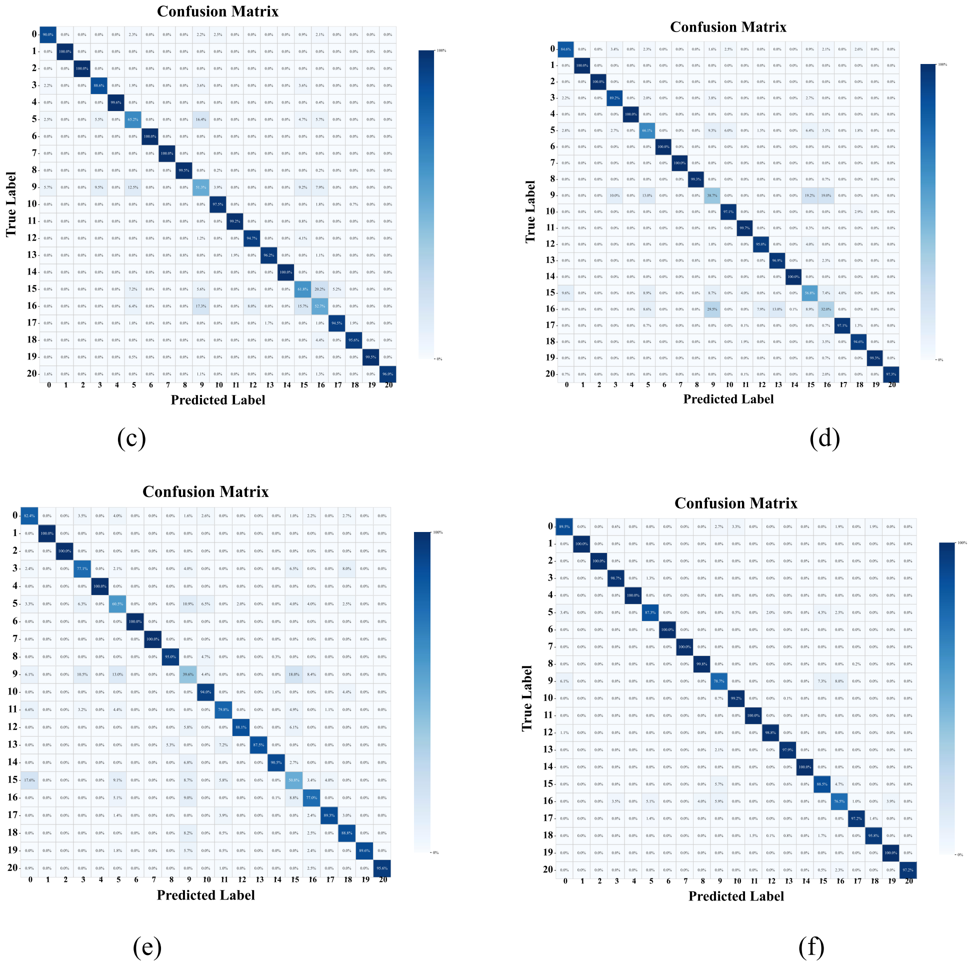


Figure 4. Confusion matrix of TE: (a)GAT, (b)TAGCN, (c)IAGNN, (d)MRF-GCN, (e)MTGNN,(f)MCTEGNN.

Figure 4 illustrates the comparative results of confusion matrices under several models, with the true categories and predicted categories listed in the rows and columns of the matrix, respectively.

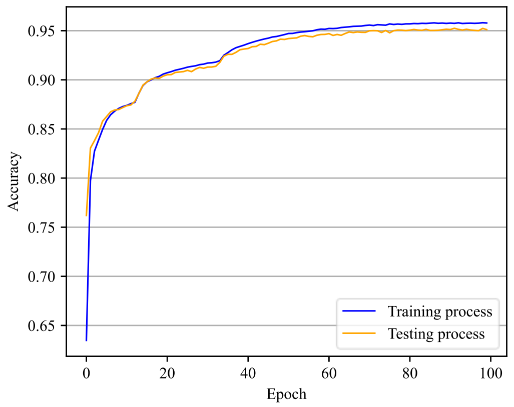


Figure 5. Classification accuracy curve of MCTEGNN on TE dataset.

Figure 5 and Figure 6 respectively illustrate the accuracy curve and training loss curve of the MCTEGNN model for fault classification on the TE dataset. The two curves not only demonstrate the performance changes during the model training process, but also reflect the stability and convergence at different training stages. Figure 7 presents a comparison of the classification accuracy F1 scores obtained through multiple experiments for several models. This comparison serves to further validate the superiority and robustness of the MCTEGNN model in the context of fault diagnosis in complex industrial processes.

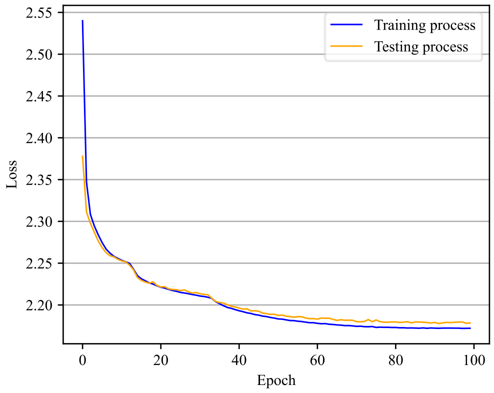


Figure 6. Training loss curve of MCTEGNN on TE dataset.

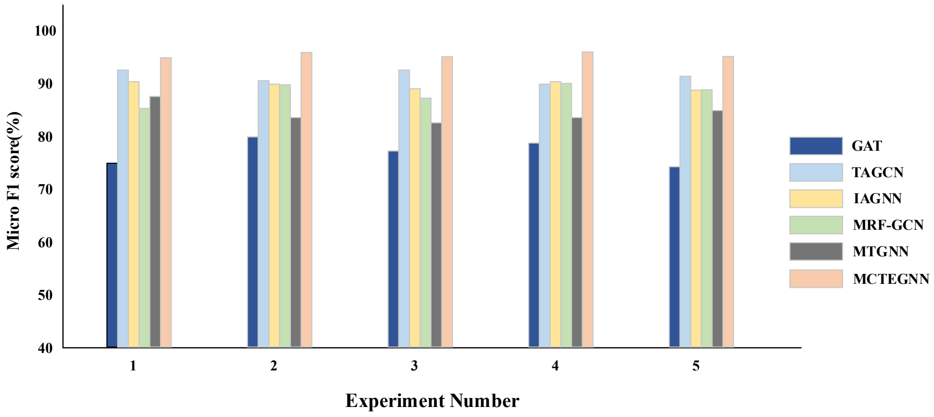


Figure 7. Comparison of Micro F1 score bar charts for different models on the TE dataset.

4.3. Classification Result Analysis of TFF Experiment

The detailed evaluation results of the TFF dataset are shown in Table 8. We perform fault diagnosis classification on the TFF dataset based on six graph-based models that can be used for industrial fault diagnosis.

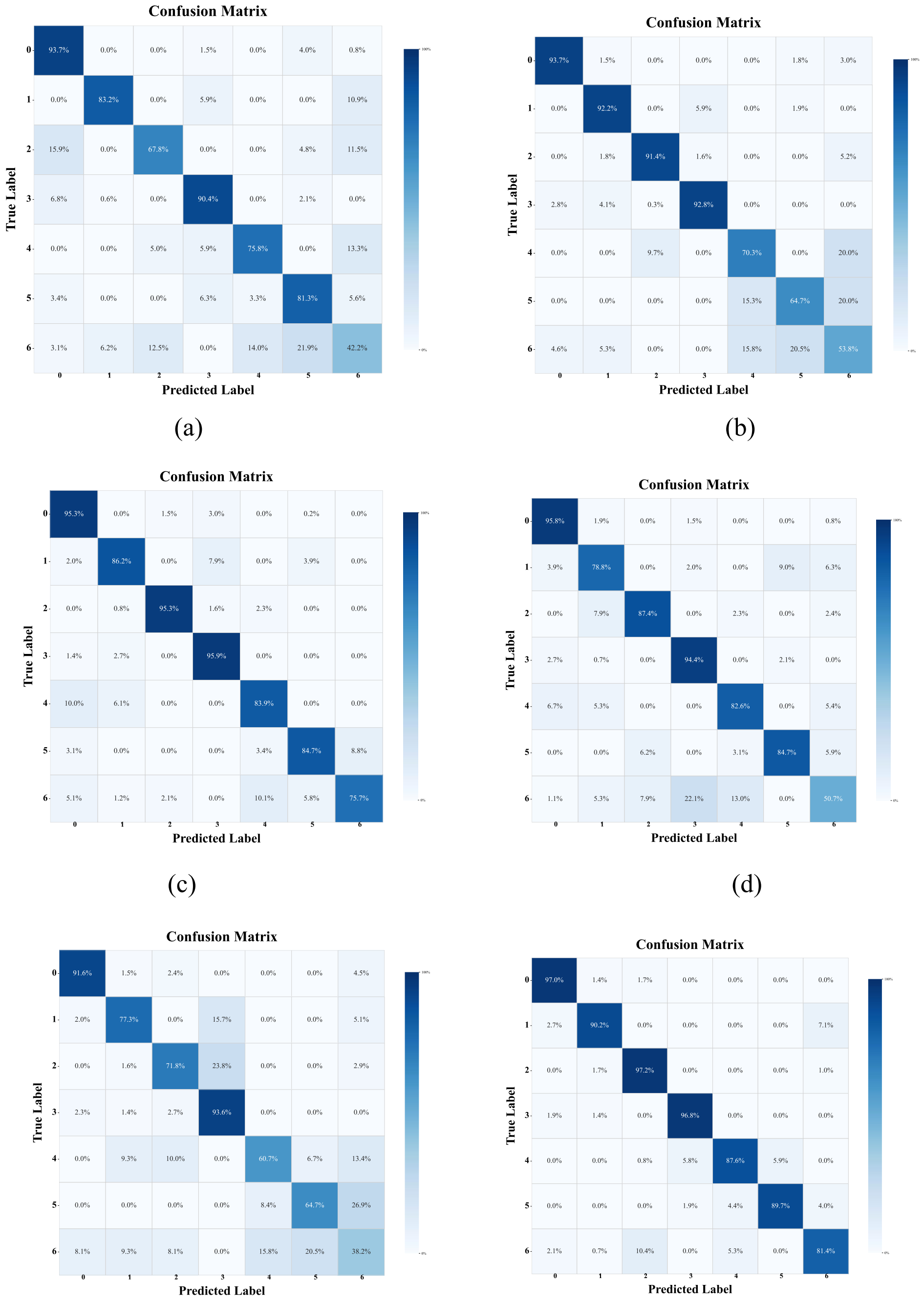
Comparing with the results of several baseline models including GAT, TAGCN, MRF-GCN, IAGNN and MTGNN, it can be observed that MCTEGNN achieved the best or comparable performance. From the classification results of 6 faults and normal states that need to be diagnosed in the TFF dataset, it can be seen that the proposed MCTEGNN achieves the best diagnostic accuracy among the six diagnostic methods. Compared with the baseline methods, the proposed MCTEGNN achieved the best classification results, with the highest micro F1 score of 0.94.77, and the macro F1 score is 0.9454. This provides further evidence of the effectiveness of the model proposed in this article.

Table 8. Classification Accuracy (%) for Various Fault Types on The TFF Dataset.

Fault types	GAT	TAGCN	IAGNN	MRF-GCN	MTGNN	**MCTEGNN**
Normal	93.74	93.8	95.32	95.84	91.63	**96.95**
Fault 1	83.24	92.2	86.23	78.77	77.27	90.17
Fault 2	67.83	91.36	95.28	87.84	71.25	**97.24**
Fault 3	90.42	92.8	95.94	9894.39	93.6	**96.77**
Fault 4	75.76	70.28	83.91	82.61	60.69	**87.57**
Fault 5	81.35	64.69	84.69	84.7	64.69	**89.69**
Fault 6	42.22	53.8	75.67	50.67	38.17	**81.43**

As shown in the comparison results in Table 8, the proposed MCTEGNN consistently outperforms the baseline model in multiple fault categories. This advantage stems from its clear consistency with the inherent coupling characteristics of industrial processes. These innovations collectively enhance the model's generalization ability under different fault modes, and their significant accuracy improvement validates this. The experiment has demonstrated the effectiveness of MCTEGNN in modeling complex industrial variable relationships for accurate fault diagnosis.

Figure 8 illustrates the comparative results of confusion matrices under several models. It can be seen from the figures that the proposed model has achieved significant improvement in classification accuracy. Figure 9 presents a comparison of the classification accuracy F1 scores obtained through multiple experiments for several models.



(a) (b)

(c) (d)

(e) (f)

Figure 8. Confusion matrix of TFF: (a)GAT, (b)TAGCN, (c)IAGNN, (d)MRF-GCN, (e)MTGNN,(f)MCTEGNN.

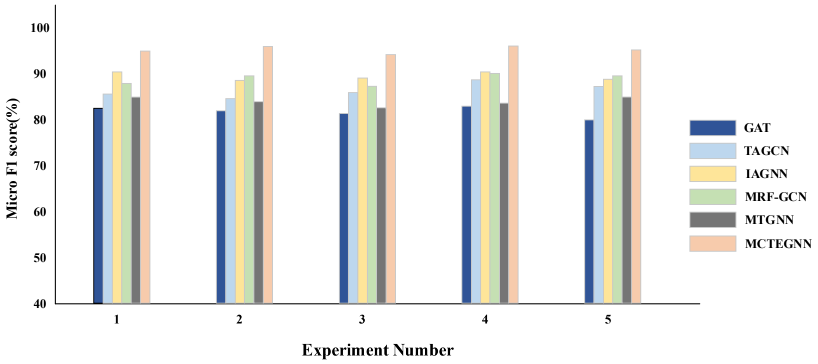


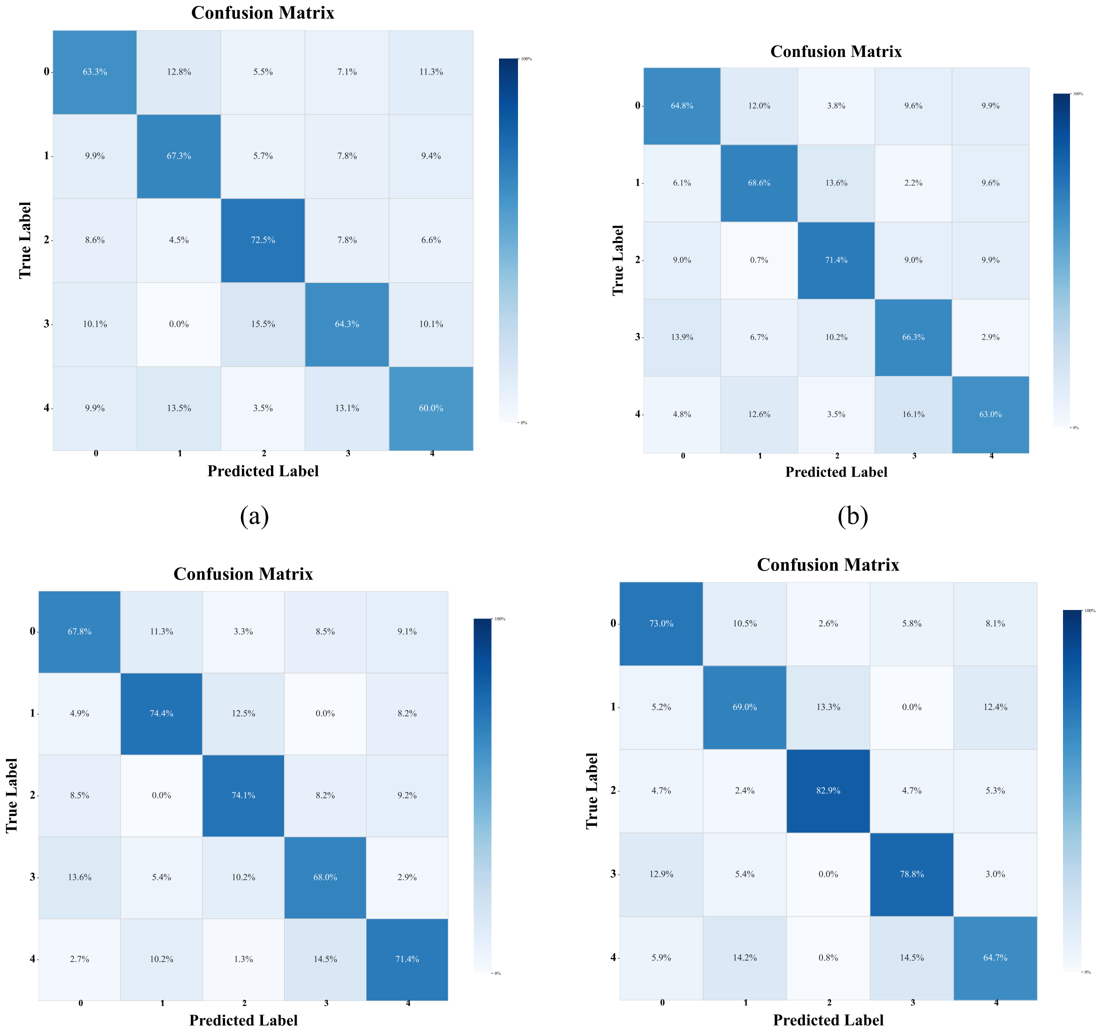
Figure 9. Comparison of Micro F1 score bar charts for different models on the TFF dataset.

4.3. Classification Result Analysis of BF Experiment

The detailed evaluation results of the BF dataset are shown in Table 9. We perform fault diagnosis classification on the BF dataset based on five graph-based models that can be used for industrial fault diagnosis. Comparing with the results of several baseline models including GAT, TAGCN, MRF-GCN and MTGNN, it can be observed that MCTEGNN achieves optimal or comparable performance on this real industrial dataset. Figure 10 illustrates the comparative results of confusion matrices under several models.

Table 9. Classification Accuracy (%) for Various Fault Types on The BF Dataset.

Fault types	GAT	TAGCN	MRF-GCN	MTGNN	**MCTEGNN**
Fault 1	63.28	64.78	67.83	73.04	**78.36**
Fault 2	67.26	68.56	74.39	69.04	**81.94**
Fault 3	72.5	71.36	74.12	82.88	**86.05**
Fault 4	64.35	66.35	67.99	78.76	**83.72**
Fault 5	60.04	63.04	71.36	64.69	**89.48**

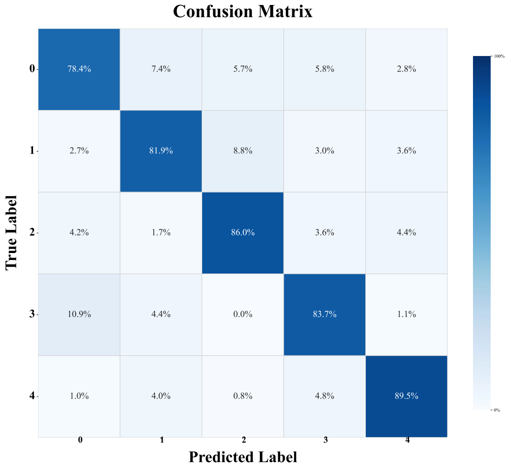


(a)

(b)

(c)

(d)



(e)

Figure 10. Confusion matrix of BF: (a)GAT, (b)TAGCN, (c)MRF-GCN, (d)MTGNN,(e)MCTEGNN.

The MCTEGNN model achieved good classification performance on the BF dataset, mainly due to its deep fit between the model structure and the complex dynamic characteristics of blast furnaces. By modeling the sensor coupling relationship across timestamps through a multi connected spatiotemporal graph (such as the lag correlation between oxygen enrichment changes and top temperature fluctuations), and quantifying the process delay effect using an attenuation matrix, the unique delay coupling mechanism of blast furnaces was effectively captured. The sliding window design of mobile pooling GNN accurately identifies local abrupt working conditions, solving the shortcomings of traditional methods in modeling cross temporal dependencies. This dynamic coupling modeling capability enables the model to accurately analyze multivariate chain faults in blast furnaces.

A decay matrix is employed to augment the graph in order to ensure an accurate representation of correlation between various sensors at different time points. The selection of decay rate δ is crucial and affects the correction of the generated spatial-temporal connectivity graph. Relevant evaluations are needed. When δ is set to 0.7, the model demonstrates satisfactory performance, which is in line with the actual interaction relationship between variables in industrial processes. When the decay rate is set to a smaller parameter, performance decreases. When δ=1, it is equivalent to the absence of a decay matrix. The proposed model is capable of learning more effective representations, and this comprehensive modeling approach has been shown to achieve excellent overall performance in industrial fault diagnosis tasks. The proposed MCTEGNN model includes multi connected graph construction and moving pool graph convolution layers. Compared to simpler, traditional models, it may need higher computational costs.

5. CONCLUSION

In order to model the comprehensive Spatial-Temporal dependencies in complex industry time signal data, we have designed a new method called Multi-Connected Temporal Encoding Graph Neural Network (MCTEGNN). A multi-connected graph is designed to connect sensors among all timestamps by additionally considering the correlations between different sensors at different timestamps. This enables comprehensive ST dependencies modeling within complex industry data. Subsequently, a multi-connected graph convolution is devised, comprising a moving-pooling GNN. This is achieved by employing a moving window and temporal pooling, which enables the capture of local ST dependencies and the subsequent learning of high-level features. The proposed model addresses the performance issues associated with inaccurate spatial-temporal modeling in complex industrial fault diagnosis. MCTEGNN has achieved significant results in modeling industrial dependencies, and its graph structure still relies on predefined decay mechanisms. Future work can combine this method with dynamic graph structures to further enhance the adaptability of the model to complex industrial scenarios. The effectiveness of our method is demonstrated through comprehensive experimentation.

Acknowledgements

This work was supported by National Natural Science Foundation of China (U21A20475), Natural Science Foundation of Hebei Province of China(E2022501017, F2020501040), Research Fund from the State Key Laboratory of Rolling and Automation, Northeastern University(2021RALKFKT007), Fundamental Research Funds for the Central Universities (N2223001).

Referrence

1. Tian Z, Zhuo M, Liu L, et al. Anomaly detection using spatial and temporal information in multivariate time series[J]. Scientific Reports, 2023, 13(1): 4400. https://doi.org/10.1038/s41598-023-31193-8
2. Wu Y, Dai H N, Tang H. Graph neural networks for anomaly detection in industrial Internet of Things[J]. IEEE Internet of Things Journal, 2021, 9(12): 9214-9231. https://doi.org/10.1109/JIOT.2021.3094295
3. Corso G, Stark H, Jegelka S, et al. Graph neural networks[J]. Nature Reviews Methods Primers, 2024, 4(1): 17. https://doi.org/10.1038/s43586-024-00294-7
4. Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(5): 4027-4035. https://doi.org/10.1609/aaai.v35i5.16523
5. Zhao Z, Li J, Liu G. Automated Graph Contrastive Learning Based on Node-Level and Edge-Level Learnable Augmentation[J]. IEEE Transactions on Computational Social Systems, 2025. https://doi.org/ 10.1109/TCSS.2025.3551250
6. Motie S, Raahemi B. Financial fraud detection using graph neural networks: A systematic review[J]. Expert Systems with Applications, 2024, 240: 122156. https://doi.org/10.1016/j.eswa.2023.122156
7. Ding C, Sun S, Zhao J. MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection[J]. Information Fusion, 2023, 89: 527-536. https://doi.org/10.1016/j.inffus.2022.08.011
8. Wu D, Zhao J. Process topology convolutional network model for chemical process fault diagnosis[J]. Process Safety and Environmental Protection, 2021, 150: 93-109. https://doi.org/10.1016/j.psep.2021.03.052
9. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arxiv preprint arxiv:1609.02907, 2016 .https://doi.org/10.48550/arXiv.1609.02907
10. Fan S, Liu G, Li J. A heterogeneous graph neural network with attribute enhancement and structure-aware attention[J]. IEEE Transactions on Computational Social Systems, 2023, 11(1): 829-838. https://doi.org/ 10.1109/TCSS.2023.3239034
11. Shin J H, Bae J, Kim J M, et al. An interpretable convolutional neural network for nuclear power plant abnormal events[J]. Applied Soft Computing, 2023, 132: 109792. https://doi.org/10.1016/j.asoc.2022.109792
12. Meng H, Geng M, Han T. Long short-term memory network with Bayesian optimization for health prognostics of lithium-ion batteries based on partial incremental capacity analysis[J]. Reliability Engineering & System Safety, 2023, 236: 109288. https://doi.org/10.1016/j.ress.2023.109288
13. Liu Z, Wan G, Prakash B A, et al. A review of graph neural networks in epidemic modeling[C]//Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024: 6577-6587. https://doi.org/10.1145/3637528.3671455
14. Jeong Y. Fault detection with confidence level evaluation for perception module of autonomous vehicles based on long short term memory and Gaussian Mixture Model[J]. Applied Soft Computing, 2023, 149: 111010. https://doi.org/10.1016/j.asoc.2023.111010
15. Deng C, Deng Z, Miao J. Semi-supervised ensemble fault diagnosis method based on adversarial decoupled auto-encoder with extremely limited labels[J]. Reliability Engineering & System Safety, 2024, 242: 109740. https://doi.org/10.1016/j.ress.2023.109740
16. Li T, Zhao Z, Sun C, et al. Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction[J]. Reliability Engineering & System Safety, 2021, 215: 107878. https://doi.org/10.1016/j.ress.2021.107878
17. Elshenawy L M, Chakour C, Mahmoud T A. Fault detection and diagnosis strategy based on k-nearest neighbors and fuzzy C-means clustering algorithm for industrial processes[J]. Journal of the Franklin institute, 2022, 359(13): 7115-7139. https://doi.org/10.1016/j.jfranklin.2022.06.022
18. Chen Z, Xu J, Alippi C, et al. Graph neural network-based fault diagnosis: A review[J]. arxiv preprint arxiv:2111.08185, 2021.https://doi.org/10.48550/arXiv.2111.08185
19. Li J, Wang Y, Zi Y, et al. Causal consistency network: A collaborative multimachine generalization method for bearing fault diagnosis[J].

IEEE Transactions on Industrial Informatics, 2022, 19(4): 5915-5924. https://doi.org/10.1109/ TII.2022.3174711

20. Zhang Z, Wu L. Graph neural network-based bearing fault diagnosis using Granger causality test[J]. Expert Systems with Applications, 2024, 242: 122827. https://doi.org/10.1016/j.eswa.2023.122827
21. Wang T, Liu Z, Lu G, et al. Temporal-spatio graph based spectrum analysis for bearing fault detection and diagnosis[J]. IEEE Transactions on Industrial Electronics, 2020, 68(3): 2598-2607. https://doi.org/10.1109/TIE.2020.2975499
22. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. IEEE transactions on neural networks and learning systems, 2020, 32(1): 4-24. https://doi.org/10.1109/TNNLS.2020.2978386
23. Li T, Zhao Z, Sun C, et al. Multi receptive field graph convolutional networks for machine fault diagnosis[J]. IEEE Transactions on Industrial Electronics, 2020, 68(12): 12739-12749. https://doi.org/10.1109/tie.2020.3040669
24. Zhang X, Gao Y, Lin J, et al. Tapnet: Multivariate time series classification with attentional prototypical network[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 6845-6852. https://doi.org/10.1609/aaai.v34i04.6165
25. Ju W, Fang Z, Gu Y, et al. A comprehensive survey on deep graph representation learning[J]. Neural Networks, 2024: 106207. https://doi.org/10.1016/j.neunet.2024.106207
26. James J Q. Graph construction for traffic prediction: A data-driven approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(9): 15015-15027. https://doi.org/10.1109/TITS.2021.3136161
27. Aggarwal K, Joty S, Fernandez-Luque L, et al. Adversarial unsupervised representation learning for activity time-series[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 834-841.https://doi.org/10.1609/aaai.v33i01.3301834
28. Yue Z, Wang Y, Duan J, et al. Ts2vec: Towards universal representation of time series[C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(8): 8980-8987. https://doi.org/10.1609/aaai.v36i8.20881
29. Gupta A, Gupta H P, Biswas B, et al. Approaches and applications of early classification of time series: A review[J]. IEEE Transactions on Artificial Intelligence, 2020, 1(1): 47-61. https://doi.org/10.1109/TAI.2020.3027279.
30. Zheng Y, Koh H Y, Jin M, et al. Graph spatiotemporal process for multivariate time series anomaly detection with missing values[J]. Information Fusion, 2024, 106: 102255. https://doi.org/10.1016/j.inffus.2024.102255
31. Liu C L, Hsaio W H, Tu Y C. Time series classification with multivariate convolutional neural network[J]. IEEE Transactions on industrial electronics, 2018, 66(6): 4788-4797. https://doi.org/10.1109/tie.2018.2864702
32. Tan Z, Zhu Y, Liu B. Learning spatial-temporal feature with graph product[J]. Signal Processing, 2023, 210: 109062. https://doi.org/10.1016/j.sigpro.2023.109062
33. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30. https://doi.org/ 10.48550/ARXIV.1706.03762
34. Wang Y, Wu M, Li X, et al. Multivariate time-series representation learning via hierarchical correlation pooling boosted graph neural network[J]. IEEE Transactions on Artificial Intelligence, 2023, 5(1): 321-333. https://doi.org/ 10.1109/TAI.2023.3241896
35. Xiao Y, Yin H, Zhang Y, et al. A dual‐stage attention‐based Conv‐LSTM network for spatio-temporal correlation and multivariate time series prediction[J]. International Journal of Intelligent Systems, 2021, 36(5): 2036-2057. https://doi.org/10.1002/int.22370
36. Yang C L, Yang C Y, Chen Z X, et al. Multivariate time series data transformation for convolutional neural network[C]//2019 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2019: 188-192. https://doi.org/10.1109/SII.2019.8700425
37. Zerveas G, Jayaraman S, Patel D, et al. A transformer-based framework for multivariate time series representation learning[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 2114-2124. https://doi.org/10.1145/3447548.3467401
38. Wang Y, Xu Y, Yang J, et al. Graph-aware contrasting for multivariate time-series classification[C]//Proceedings of the AAAI conference on artificial intelligence. 2024, 38(14): 15725-15734. https://doi.org/10.1609/aaai.v38i14.29501
39. Van der Maaten L, Hinton G. Visualizing data using t-SNE[J]. Journal of machine learning research, 2008, 9(11).
40. Bathelt A, Ricker N L, Jelali M. Revision of the Tennessee Eastman process model[J]. IFAC-PapersOnLine, 2015, 48(8): 309-314. https://doi.org/10.1016/j.ifacol.2015.08.199
41. Ruiz-Cárcel C, Cao Y, Mba D, et al. Statistical process monitoring of a multiphase flow facility[J]. Control Engineering Practice, 2015, 42: 74-88. https://doi.org/10.1016/j.conengprac.2015.04.012

42. Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks[J]. stat, 2017, 1050(20): 10-48550.https://doi.org/10.48550/arXiv.1710.10903
43. Du J, Zhang S, Wu G, et al. Topology adaptive graph convolutional networks[J]. arxiv preprint arxiv:1710.10370, 2017. https://doi.org/10.48550/arXiv.1710.10370
44. Chen D, Liu R, Hu Q, et al. Interaction-aware graph neural networks for fault diagnosis of complex industrial processes[J]. IEEE Transactions on neural networks and learning systems, 2021, 34(9): 6015-6028. https://doi.org/10.1109/TNNLS.2021.3132376
45. Wu Z, Pan S, Long G, et al. Connecting the dots: Multivariate time series forecasting with graph neural networks[C]//Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020: 753-763. https://doi.org/10.1145/3394486.3403118