Graph Attention Convolutional Neural Network combined with Transformer Classification Method for Small Sample Datasets

Rongzhen Lei[1], Fan Yang[2,*] and Yanhui Ren[2]

[1] School of Instrument Science and Opto-Electronics Engineering, Beijing Information Science and Technology University, Beijing 100192, China;
[2] Department of Automation, Tsinghua University, Beijing 10084, China

Abstract. In view of the shortcomings of deep learning methods in modeling small sample fault data, this paper proposes a fault classification method based on Graph Attention Convolutional Neural Network (GATCN) combined with Transformer, which can be well adapted to small sample fault data sets. Firstly, the multi-dimensional sensor data is converted into a multi-component graph representation, and the weights of the edges are updated by the attention mechanism between the nodes of the graph to represent the topological structure between different nodes, to extract the correlation between the multi-component parameters. Then, the Transformer module learn the extracted spatial features to capture the temporal feature relationships of the time series data. Finally, the classification layer based on Softmax is used to classify faults. The model is demonstrated on the TEP data set and the fault datasets from three satellite subsystems. The results show that our model is superior for small sample fault data sets.

Keywords: Small Sample, Fault Classification, Graph Attention Neural Network, Graph Convolutional Neural Network, Transformer.

1 Introduction

With the increasing complexity and intelligence of engineering equipment, fault classification technology has become one of the key technologies to ensure the safe and stable operation of equipment. Traditional fault classification methods mainly rely on human experience and physical models [1], but these methods often have problems such as low efficiency, insufficient accuracy, and difficulty in modeling. In recent years, with the improvement of sensing and computing technology, more and more on-site real-time data are available, which creates favorable conditions for fault classification methods based on data-driven deep learning [2]. However, in many scenarios, there are only small samples and unbalanced fault data, and general deep learning methods cannot effectively extract data features, resulting in low diagnostic accuracy [3]. Graph Attention Neural Network (GAT) [4] and Transformer [5], as two powerful deep learning models, show great potential in the field of fault classification.

Fault classification methods are divided into knowledge-based and data-driven. Knowledge-based methods rely on models and expertise but falter in complex systems.In data-driven deep learning, models learn useful feature representations from data with minimal reliance on prior knowledge, thereby enhancing their generalization capabilities. This enables them to handle high-dimensional and complex datasets, uncover hidden patterns and rules within the data, and effectively perform fault classification.

One of the existing data-driven deep learning fault classification methods is Convolutional Neural Network (CNN) [6-8], The architecture of CNNs, designed with convolutional layers for applying filters to local input data and followed by pooling layers for dimensionality reduction, adeptly captures spatial features from images but has limitations in capturing temporal dynamics or long-term dependencies in time series data, which may lead to overlooking important temporal features critical for fault detection in systems where time-dependent patterns play a significant role. Recurrent Neural Network (RNN), with their architecture of recurrent layers that maintain hidden states across time steps, can recognize and predict failure modes by learning time dependencies in data. However, for long time series, RNNs cannot effectively capture all dependencies [9-10], and their inference speed is limited, they are also sensitive to input noise and outliers [11].

Deep Belief Networks (DBNs) are composed of multiple layers of Restricted Boltzmann Machines (RBMs) topped with a final softmax classification layer, utilizing an unsupervised pre-training phase followed by supervised fine-tuning [12]. However, this complex architecture is prone to overfitting, particularly on small datasets, as the unsupervised pre-training might not efficiently capture fault-specific features when data is limited. Consequently, this can lead to extended training times [13-14]and poor generalization to small sample sizes, rendering DBNs less effective for real-time fault detection tasks. In addition, the above methods all have high requirements on the amount of data. For the small sample fault, feature extraction is incomplete, and the deep representation of the data cannot be learned [15].

The powerful graph data processing capability of semi-supervised Graph Convolutional Neural Network (GCN) enables it to effectively process complex graph-structured data in fault classification. In complex systems, the connection relationship between devices can be naturally represented as graph structure [16]. GCN can use this graph structure data to extract fault features, including node features, edge features, and connection relationships between nodes, through information transmission and aggregation among nodes, to comprehensively capture fault features in the system [17]. However, the content of node information that GCN can capture is too small to obtain the global node relationship, while GAT will selectively aggregate information from neighbor nodes through the attention mechanism when transmitting messages, without using any matrix operation to select the information of neighbor nodes. Thus, the expression capability of graph convolutional networks is greatly improved [18].

By integrating Graph Convolutional Network (GCN) with attention mechanisms, we have developed a Graph Attention Convolutional Neural Network (GATCN) that not only establishes a feature graph of effective fault data but also fully utilizes known label data to extract the relationships between nodes and edges within the graph structure, without an over-reliance on data quantity accumulation. To enable the model to learn both temporal and spatial correlations among parameters, we have combined the GATCN with the Transformer's [19] temporal feature extraction capabilities, resulting in an enhanced model known as GATCN-Transformer. This hybrid architecture allows GATCN to handle spatial features using graph-based attention, while the Transformer processes sequential temporal data using self-attention mechanisms across time steps. The GATCN-Transformer leverages the spatial feature extraction capabilities of GATCN and the temporal feature extraction strengths of Transformers, with the Transformer's self-attention mechanism adept at capturing long-term dependencies and dynamic patterns in time series data, complementing GATCN's graph-based feature extraction. This approach addresses the limitations of CNNs, RNNs, and DBNs by providing a more holistic method that excels in both spatial and temporal dimensions, offering enhanced accuracy and real-time fault detection capabilities, especially effective with small datasets and dynamic data. The advantages of GATCN-Transformer have been experimentally verified through comparisons with other methods, and its effectiveness for small sample faults was validated by augmenting the small sample fault dataset and comparing the accuracy of different models under varying data sizes.

2 Method

In this paper, the classification of time series faults occurring in real industrial processes is transformed into the classification of graphs. As shown in Figure 1, we convert the multidimensional time series into a multivariate graph $G(V,E,A)$, which mainly includes node V, edge E and adjacency matrix $\boldsymbol{A} \in \mathrm{R}^{\mathrm{N\times N}}$. Each node V represents a one-dimensional variable, n-dimensional time series data contains N nodes, and the node matrix $\boldsymbol{X} = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{N}\}$, where $X_i \in \mathrm{R}^{\mathrm{T}}$, which contains all node information. All nodes are connected through edge E, and each edge is assigned a different weight vector, representing the connection relationship between different nodes, which is represented by the adjacency matrix $\boldsymbol{A}$. Then, the constructed multivariate graph G is computed with attention, and the feature information between different variables is extracted. Input the calculated results into Transformer for reconstruction and output, so that the obtained feature graph contains both spatial and temporal features. Finally, the fault classification is carried out through Softmax, and the fault classification task is transformed into the classification task of different graphs.

The model comprises three parts: a Graph Attention Neural Network for spatial feature extraction, a Transformer for temporal feature extraction, and a residual-based fault classification layer. Multi-dimensional time series are segmented into subsequences that form multivariate time graphs. Features are extracted using a graph convolutional network to determine node connections. These features are input into the Transformer, processed through multi-head attention and residual connections to capture spatiotemporal correlations. A fully connected layer maps these features, and a SoftMax layer classifies faults. Details of each module are elaborated in subsequent sections.

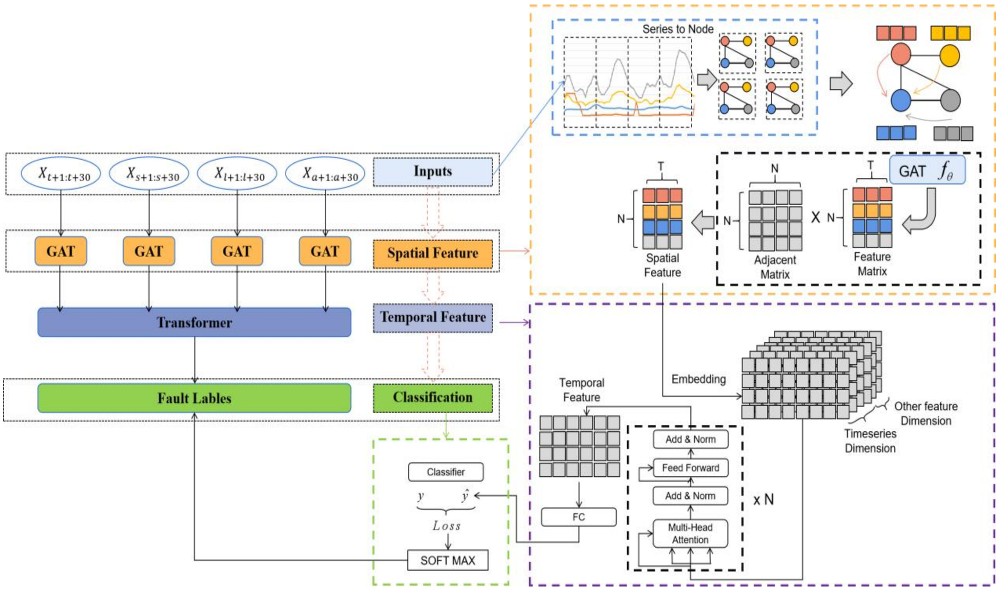

Fig. 1. Model structure diagram.

2.1 The spatial feature extraction layer of a graph convolutional neural network based on attention

Since the time series itself does not have a graph structure directly, it needs to be transformed first. As shown in Figure 2, for multidimensional time series, each one-dimensional parameter is regarded as a node in the time graph, and the node matrix $\boldsymbol{X}$ is obtained by selecting an appropriate time window. Then initialize the connection between the nodes to get the edge set $E = \{\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_\mathrm{N}\}$, In set E, there can be at most $N^2 \times 2$ edges, the first column is the starting point of the edge, the second column is the end point, and the adjacency matrix $\boldsymbol{A}$ of the edge represents the weight of each edge [20].

We used GAT to extract spatial features from the original time series [21]. For a single node h_i, the calculation process of its attention coefficient is as follows:

$$s_{ij} = LeakyReLU\left(\vec{a}^T\left[W\vec{h}_i \parallel W\vec{h}_j\right]\right) \tag{1}$$

$$a =< h_i, h_j >/\sqrt{N} \tag{2}$$

This represents the importance of the features of node j to node i. a is the attention function, <> represents the dot product. Then all the obtained attention scores are normalized, and the calculation process is as follows:

$$\alpha_{ij} = softmax\left(s_{ij}\right)\frac{exp\left(s_{ij}\right)}{\sum_{k\in N_i} exp\,(s_{ik})} \tag{3}$$

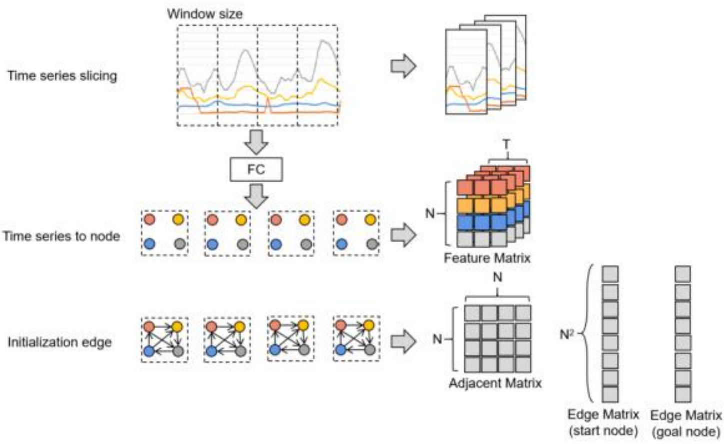

Fig. 2. Transforms the time series into a graph structure.

The weighted sum results in the attention values of all other nodes to node i, as shown in Formula 4. At the same time, in order to stabilize the learning process of attention, the mechanism of multi-head attention is adopted, and the number of heads is selected as k [22], as shown in Equation 5.

$$\alpha_i^{'} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\, Wh_j\right) \tag{4}$$

$$\alpha_i^{'} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K} \sum_{j\in N_i} \alpha_{ij}^{k}\, W^k\vec{h}_j\right) \tag{5}$$

Then the model output is updated using graph convolution [23].

$$output = \tilde{L}_{sym}XW \tag{6}$$

$$\tilde{L}_{sym} = \tilde{A} \odot M \tag{7}$$

$$\tilde{A} = A + I \tag{8}$$

where X is the node matrix, W represents the projection matrix used to enhance the node feature and obtain more spatial representation. Matrix I is the identity matrix, representing the node's connection to itself. Matrix M is the attention matrix of all nodes.

$$M = Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)(V) \tag{9}$$

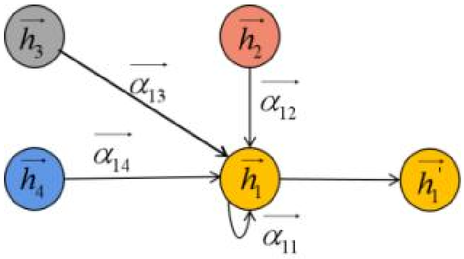

Fig. 3. Figure node attention calculation process.

2.2 Time feature extraction layer based on Transformer

The node feature matrix X' output by GAT is taken as the input of Transformer [24]. The feature matrix first passes through the embedding layer for positional encoding to get suitable input for the Transformer.

$$X^{'} = embed(X) \tag{10}$$

The Transformer's encoder extracts temporal features and captures dependencies across time steps via multi-head attention. It is followed by a feed-forward network, residual connections, and layer normalization to enhance training stability and convergence. A fully connected layer then maps the output to target dimensions, enabling the model to learn from minor input variations while maintaining feature distribution stability.

$$X^{'} = LayerNorm\left(X^{'} + MultiHeadAttention(X^{'})\right) \tag{11}$$

$$X^{\quad '} = LayerNorm\left(X^{\quad '} + FeedForward\left(X^{\quad '}\right)\right) \tag{12}$$

$$y = tanh(\alpha X^{'}\Theta) \tag{13}$$

In this way, the output features through Transformer contain both the previous spatial and temporal correlation features. Finally, the raw output is transformed into a probability distribution through the Softmax layer, which describes the probability of the input data belonging to each category.

3 Experiments and Discussions

The software environment of the experiment is Windows11 64-bit operating system, with 6GB of RAM, using the PyTorch framework on the PyCharm platform. The hardware environment is an AMD Ryzen 75800H processor and an RTX 3060 Laptop GPU graphics card. The model parameters are set as follows:

(1) Using the Adam optimizer, with the initial learning rate set to the default value of 0.001;

(2) The learning rate attenuation mechanism is adopted, and the learning rate is multiplied by the attenuation coefficient 0.1 for every 5 epochs of training, so as to solve some unstable situations and improve the stability of the model.

3.1 Data set and experimental setup

The experimental data was selected as Tennessee Eastman Process Data Set (TEP), which is a practical chemical process simulation data set and a classical data set used for anomaly detection and process control, with relatively few fault data samples, belonging to a typical small sample data set. It contains 33 process variables and 19 quality variables. The data set covers 21 fault conditions and one normal condition, each containing a training sample and a test sample. The training samples were obtained under 25h running simulation, and the total number of observations was 500. The sample of the test set was obtained under 48h simulation, with the fault introduced at the 8th hour. A total of 960 observed values were collected, among which the first 160 observed values were normal data, and the data sampling time was 3 minutes. In the experiment, all fault samples were selected, a total of 21 fault types, each of which contained all 52 parameters, and the model was trained using training set. At the same time, the simulated in-orbit satellite subsystem data were selected for the experiment, namely, attitude and orbit control subsystem (AOCS), power supply and distribution subsystem (PSD) and laser payload subsystem (LPS). The AOCS contains 74 parameters and 13 types of faults, PSD contains 51 parameters, 13 types of faults, and LPS contains 126 parameters, 13 type of faults. At the same time, all the faults in the four data sets are divided into two types: slowly-varying faults and abrupt faults. Table 1 provides detailed descriptions of the different datasets. The second column shows the total amount of original fault data, the third column presents the total amount of data after sliding window processing, and the last column lists the number of fault types in different datasets.

Table 1. Data set information.

Data set	Raw data length	Total amount of data after conversion	Number of fault types
TEP	17010,52	1680,30,52	21
PSD	35950,51	3548,30,51	13
LPS	45860,126	4547,30,126	13
AOCS	26890,74	2650,30,74	13

3.2 Time window selection

In data analysis and model construction, the choice of time window determines the data features and patterns that can be captured by the model. Larger time windows can capture longer-term trends in the data, reducing biases due to random perturbations and providing a better global view. However, a small time window can quickly respond to short-term changes and has higher flexibility. Considering the sampling frequencies of four data sets, 30 is selected as the time window of the model, which can not only ensure that the data has sufficient temporal information, but also meet the requirements of rapid response of the diagnostic model.

3.3 Case Studies and Experimental Results Analysis

When training the model using the time window transformed data, it should be noted that the data were not entered in time series order. This is due to the particularity of the TEP data set. Compared with the satellite subsystem data set, the fault representation mode of the TEP data set is more complex, and the vibration is obvious when the fault occurs, as shown in Figure 4. In addition, TEP data sets have long failure periods and relatively few data points. Therefore, in the actual process of data acquisition, we adopted a ratio of 7:2:1 for random data acquisition for experimental verification.

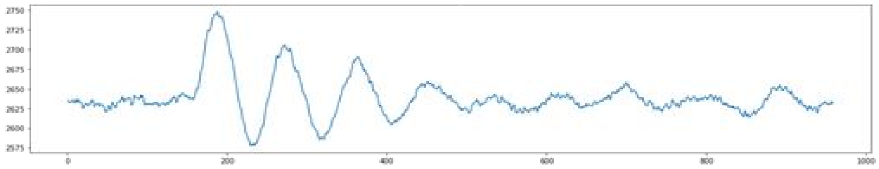

Fig. 4. TEP fault mode.

Case 1: Classification based on global fault mode. Accuracy, precision, recall and F1 score are selected as our evaluation indicators, and the experimental results are shown in Figure 5. Transformer, GRU [25], $\mathrm{DGM}^2 - \mathrm{O}$ [26], and MTGNN [27] are selected for comparison. In the four data sets, our model achieves the optimal results in most indicators. Especially in the TEP data set, our model significantly outperforms other models, with an accuracy increase of 12 percentage points compared to the strongest comparison model Transformer, and the accuracy has increased from 40.7% to 93.8% with a surge of 130% compared to the $\mathrm{DGM}^2 - \mathrm{O}$ model. This significant improvement further confirms the excellent diagnostic performance of our model in the face of small sample sizes and complex failure modes. In the satellite subsystem datasets, although the comparison models has improved its performance compared to the TEP data set, the performance gap is still significant compared to our model. In PSD and LPS datasets, our model continues to lead the way. In the AOCS data set, the Transformer model is only 1 percentage point more accurate than our model. The explanation for this difference is that compared with the fault data in the TEP data set, the fault mode of the satellite subsystem is relatively simple, which enables the comparison model to achieve better performance in the training process, resulting in different performance trends in the two data sets.

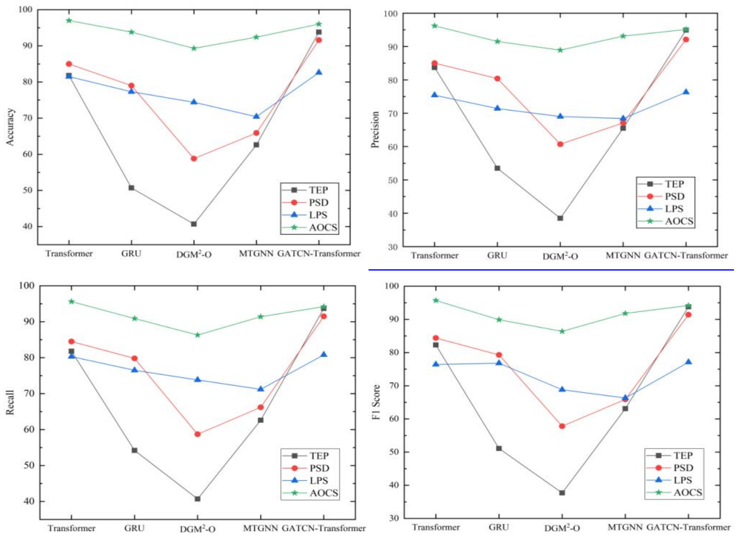

Fig. 5. Performance of the improved model and the comparison models across different metrics based on four datasets.

Table 2. Performance of the improved model and the comparison models across different metrics based on four datasets.

Methods	TEP				PSD			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Transformer	81.8±2.0	83.7±1.6	81.8±2.0	82.3±1.9	85.0±1.5	85.0±1.6	84.5±1.5	84.4±1.3
GRU	50.7±5.5	53.5±7.0	54.2±4.5	51.1±5.7	79.0±3.9	80.4±2.9	79.8±3.4	79.3±3.8
DGM^2-O	40.7±5.0	38.5±6.7	40.7±5.0	37.7±5.7	58.8±6.3	60.7±5.7	58.7±5.9	57.8±5.6
MTGNN	62.6±4.4	65.5±4.0	62.6±4.4	63.1±4.2	65.9±2.4	67.1±3.4	66.2±2.5	65.9±2.7
GATCN-Transformer	**93.8±1.9**	**94.9±1.5**	**93.7±1.9**	**93.8±1.9**	**91.6±1.9**	**92.1±2.0**	**91.5±2.1**	**91.4±2.0**
Methods	**LPS**				**AOCS**			
Transformer	81.5±1.2	75.4±0.5	80.3±0.7	76.4±0.7	**97.0±1.0**	**96.2±1.4**	**95.6±1.4**	**95.7±1.5**
GRU	77.3±3.8	71.4±3.6	76.5±3.4	76.8±3.5	93.8±0.7	91.5±0.6	90.9±0.4	89.9±1.3
DGM^2-O	74.4±4.3	69.0±4.9	73.8±2.8	68.8±4.2	89.3±4.3	88.9±4.9	86.3±4.8	86.4±5.2
MTGNN	70.4±3.7	68.4±2.9	71.2±2.5	66.3±3.1	92.4±2.4	93.1±2.1	91.4±2.4	91.8±2.4
GATCN-Transformer	**82.6±0.8**	**76.3±0.3**	**80.8±0.7**	**77.1 ±0.7**	96.0±1.2	95.1±1.5	94.2±1.8	94.2±1.8

Case 2: Classification based on different fault types. According to the actual situation, the fault types of all data sets are subdivided into abrupt faults and slowly-varying faults to evaluate the diagnostic efficiency of the model for different fault types. The experimental results show that our model shows high diagnostic performance for both fault types on most datasets. Especially in the case of more complex failure modes, our model still maintains a significant advantage over the comparison model, and in the TEP data set, our model has an accuracy advantage of 2.5 percentage points compared to the strongest comparison model Transformer. In AOCS data sets with relatively simple failure modes, our model performs equally with the comparison model on various indices. When comparing the diagnostic performance of abrupt faults and slowly-varying faults, we find that the overall performance of abrupt faults is more stable and robust. The explanation of this phenomenon is that the characteristics of abrupt faults change significantly and rapidly in a very short period of time, while the time sensitivity of slow faults is low, and there is a large time delay between the occurrence of faults and the change of fault mode, resulting in the overall performance of slow faults showing greater volatility than that of abrupt faults. Despite these challenges, our model has the best overall performance on both fault types. This result highlights the validity and reliability of our model in dealing with the dynamic characteristics of different faults.

Table 3. The experimental results for slowly-varying faults and abrupt faults in the TEP.

Methods	**TEP(abrupt faults)**				**TEP(slowly-varying faults)**			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Transformer	94.7±1.5	95.1±1.3	94.7±1.5	94.6±1.6	87.5±4.4	89.1±3.8	87.5±4.4	87.2±4.6
GRU	76.2±5.3	73.3±6.9	75.1±5.6	72.8±6.6	36.8±3.7	32.8±5.2	35.9±2.7	29.0±2.7
DGM^2-O	55.5±4.7	56.3±6.5	55.5±4.7	53.2±4.9	40.0±6.8	42.7±5.6	40.0±6.8	39.5±6.5
MTGNN	78.3±6.1	81.2±6.5	78.3±6.1	77.7±6.0	63.0±8.5	68.4±7.1	63.0±8.5	62.8±9.1
GATCN-Transformer	**97.2±1.1**	**97.6±1.0**	**97.2±1.1**	**97.2±1.2**	**91.8±4.2**	**92.7±3.9**	**91.8±4.2**	**91.7±4.3**

Table 4. The experimental results for slowly-varying faults and abrupt faults in the PSD.

Methods	**PSD(abrupt faults)**				**PSD(slowly-varying faults)**			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Transformer	90.8±2.8	90.7±2.5	90.1±2.5	90.1±2.6	93.7±1.6	93.6±1.7	93.7±1.8	93.6±1.7
GRU	85.5±2.3	86.0±2.4	85.6±2.2	85.3±2.3	91.8±1.2	91.2±1.7	91.2±1.8	91.0±1.5
DGM^2-O	70.5±8.4	71.3±8.5	71.0±8.0	69.6±9.0	82.6±4.5	83.8±4.7	82.2±4.7	81.8±4.9
MTGNN	66.7±7.7	68.0±7.6	67.5±7.4	66.6±7.7	72.9±9.3	73.5±9.9	72.5±9.2	71.7±9.7
GATCN-Transformer	**94.0±2.1**	**94.4±1.8**	**93.9±2.0**	**93.9±2.1**	**94.6±1.8**	**94.7±1.7**	**94.4±2.0**	**94.3±1.9**

Table 5. The experimental results for slowly-varying faults and abrupt faults in the LPS.

Methods	**LPS(abrupt faults)**				**LPS(slowly-varying faults)**			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Transformer	92.1±1.2	89.6±4.2	88.8±1.6	85.5±2.9	58.0±2.8	**55.0±4.0**	58.0±2.6	53.3±4.3
GRU	88.5±4.2	79.1±2.9	79.1±2.9	80.3±4.1	46.1±4.2	34.9±3.6	47.5±3.3	37.5±3.7
DGM^2-O	77.2±2.8	62.9±3.4	74.6±3.5	66.3±3.6	37.5±3.7	31.1±2.1	42.3±3.1	32.3±2.6
MTGNN	89.6±5.9	90.5±7.0	89.6±5.0	87.6±6.6	50.2±4.4	46.4±7.2	52.0±3.9	44.0±5.5
GATCN-Transformer	**94.0±1.8**	**94.2±1.5**	**94.4±1.5**	**94.2±1.6**	**62.3±2.1**	53.1±4.7	**61.7±1.7**	**56.0±2.1**

Table 6. The experimental results for slowly-varying faults and abrupt faults in the AOCS.

Methods	**AOCS(abrupt faults)**				**AOCS(slowly-varying faults)**			
	Accuracy	**Precision**	**Recall**	**F1 score**	**Accuracy**	**Precision**	**Recall**	**F1 score**
Transformer	94.7±1.6	94.7±1.9	**94.5±1.7**	**94.5±1.7**	**99.9±0.3**	**99.8±0.5**	**99.8±0.5**	**99.7±0.5**

GRU	87.6±2.1	86.6±2.5	85.4±3.0	85.4±2.6	98.3±2.7	98.5±2.2	98.9±2.7	98.2±2.7
DGM^2-O	91.4±1.8	91.0±2.3	90.3±2.1	90.0±2.2	98.6±2.5	99.1±1.5	98.9±2.0	98.9±2.0
MTGNN	92.7±3.8	92.6±4.8	91.6±4.8	91.7±4.8	99.8±0.5	99.7±0.8	99.6±0.9	99.6±0.9
GATCN-Transformer	**94.6±1.2**	**95.2±1.1**	94.4±1.3	94.2±1.5	99.8±0.4	**99.8±0.5**	99.7±0.6	**99.7±0.5**

Case 3: Performance comparison of the improved model and the comparison model with different sample sizes. In order to discuss the impact of data set size on different models, the fault samples were expanded on TEP data set using the data augmentation method [28], and the effects of the model under different data samples were discussed, as shown in Table 7 and Figure 6. It can be seen that the data sample size improves the performance of different models to different degrees. When the data set size reaches 1,500, the effect improvement of these models is no longer obvious. The diagnostic accuracy of our model in TEP data set without data expansion is 93.8%. For Transformer, our model's performance can be approached when the data sample size is doubled. However, for other comparison models, when the number of data samples reaches 1,500, the improvement in model performance almost stops, and the diagnostic accuracy is much lower than our model. It can be inferred that our model still has a good diagnostic accuracy even in the case of small samples, which is highly advantageous in practical application scenarios and can make up for the defects of small samples of fault data in actual production operations.The improved model shows significant potential for application in small-sample complex industrial systems. The model has a relatively high time complexity of $O(m \times n)$, where m represents the node features selected during graph network construction, and n represents the total node features. The hardware requirements are not very high, with 6GB of GPU memory, single card, and a runtime of under 2 hours for 30 epochs. The larger the parameter dimensions, the longer the time required. Detailed information about the dataset is provided in Table 1.

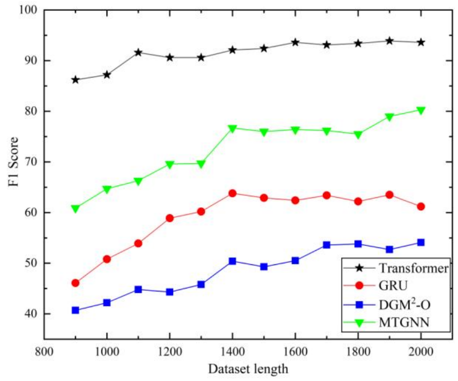

Fig. 6. Performance of the comparison model in different orders of data.

Table 7. Experimental results of the model after data augmentation.

Sample size	TEP	Accuracy	Precision	Recall	F1 score
900	Transformer	86.3±3.2	87.6±3.0	86.3±3.2	86.2±3.1
	GRU	48.9±2.8	48.8±4.6	48.6±4.5	46.1±3.0
	DGM^2-O	42.8±4.9	43.0±6.4	42.8±4.9	40.7±5.5
	MTGNN	61.2±3.6	62.9±3.5	61.2±3.6	60.9±3.6
1000	Transformer	87.3±3.3	88.3±3.3	87.3±3.3	87.2±3.3
	GRU	53.9±2.9	54.2±2.7	53.4±2.5	50.8±2.3
	DGM^2-O	45.0±5.3	44.4±7.0	45.0±5.3	42.2±6.5
	MTGNN	64.9±3.9	66.8±3.5	64.9±3.9	64.7±3.5
1100	Transformer	91.7±2.7	92.3±2.7	91.7±2.7	91.0±2.9
	GRU	57.6±3.2	56.0±6.4	56.6±5.1	53.9±5.9
	DGM^2-O	47.2±5.3	46.2±7.7	47.2±5.3	44.8±6.3
	MTGNN	66.6±7.6	68.4±8.0	66.6±7.6	66.3±7.7
1200	Transformer	90.5±3.3	91.5±2.8	90.5±3.3	90.6±3.3
	GRU	59.6±5.0	61.8±7.0	59.8±6.2	58.9±6.4
	DGM^2-O	46.7±4.4	47.1±4.6	46.7±4.4	44.3±4.7
	MTGNN	69.5±8.6	72.7±7.9	69.5±8.6	69.6±8.3
1300	Transformer	90.6±4.1	91.4±3.8	90.6±4.1	90.6±4.0
	GRU	63.8±2.4	62.8±2.9	61.9±2.0	60.2±2.7
	DGM^2-O	48.1±5.3	49.7±4.6	48.1±5.3	45.8±5.5
	MTGNN	69.5±3.3	72.5±3.3	69.5±3.3	69.7±3.3
1400	Transformer	92.2±2.1	92.8±2.1	92.2±2.1	92.1±2.2
	GRU	64.9±2.2	66.1±4.2	65.0±1.4	63.8±2.1
	DGM^2-O	51.8±5.5	53.1±6.8	51.8±5.5	50.4±6.1
	MTGNN	76.6±4.3	78.4±4.4	76.6±4.3	76.7±4.3
1500	Transformer	92.4±2.0	93.1±1.9	92.4±2.0	92.4±2.0
	GRU	63.8±2.4	64.8±1.0	64.1±1.5	62.9±1.5
	DGM^2-O	51.1±5.9	52.0±7.0	51.1±5.9	49.3±6.7
	MTGNN	75.9±3.9	78.3±2.6	75.9±3.9	76.0±3.6
1600	Transformer	93.6±2.8	94.1±2.7	93.6±2.8	93.6±2.8
	GRU	64.8±2.2	65.6±2.2	63.5±1.2	62.4±1.9
	DGM^2-O	52.5±6.8	52.2±7.6	52.5±6.8	50.5±7.8
	MTGNN	76.0±5.7	78.0±4.9	76.0±5.7	76.4±5.5
1700	Transformer	93.1±1.8	93.7±1.7	93.1±1.8	93.1±1.8
	GRU	64.8±2.4	66.3±3.7	64.4±2.8	63.4±2.8
	DGM^2-O	55.3±6.8	57.1±8.0	55.3±6.8	53.6±7.4
	MTGNN	75.9±5.3	78.2±5.4	75.9±5.3	76.2±5.2
1800	Transformer	93.4±3.9	94.0±3.5	93.4±3.9	93.4±3.9
	GRU	63.2±3.4	64.5±3.0	62.7±3.4	62.2±3.9
	DGM^2-O	55.2±6.2	56.8±5.9	55.2±6.2	53.8±6.4
	MTGNN	75.3±6.0	77.1±4.8	75.3±6.0	75.5±5.7
1900	Transformer	93.9±2.3	94.6±2.1	93.9±2.3	93.9±2.3
	GRU	64.7±3.0	65.2±2.3	64.8±2.0	63.5±2.0

	DGM^2-O	54.6±7.1	55.7±8.1	54.6±7.1	52.7±8.2
	MTGNN	78.9±4.1	80.1±3.5	78.9±4.1	79.0±4.0
2000	Transformer	93.6±2.3	94.1±2.1	93.6±2.3	93.6±2.2
	GRU	62.6±2.2	64.0±2.4	61.7±1.5	61.2±2.0
	DGM^2-O	55.4±7.8	57.0±8.0	55.4±7.8	54.1±8.7
	MTGNN	80.2±5.8	81.0±5.3	80.2±5.8	80.3±5.6

4 Conclusion

Aiming to address the problem of small sample fault classification, we propose a classification method based on GATCN-Transformer. The model structure consists of three parts, which are spatial feature extraction layer based on Graph Attention Convolutional Neural Network, temporal feature extraction layer based on Transformer and fault classification layer based on Softmax.

The datasets for the model experiments include the TEP data set and data from three satellite subsystems. The TEP data set is characterized by a small number of fault samples and complex fault patterns, making it a typical small-sample, high-dimensional, and high-complexity data sets in chemical systems. Experimental data from Case 1 show that the improved model's accuracy on the TEP data set has significantly increased (from 40.7% in the worst case for the comparison model to 93.8%), demonstrating the model's robustness in handling small sample data. Furthermore, experimental data from Case 2 indicate that the improved model consistently performs optimally on all datasets, including but not limited to the TEP data set, for both abrupt and gradual faults. This capability suggests that the model does not rely on large amounts of data to make accurate predictions but can effectively learn from limited samples, which is particularly important in data-constrained applications. The satellite subsystem datasets are characterized by high dimensionality and numerous simulated variables. Analysis of the given results shows that the improved model consistently achieves optimal performance in most cases. In the AOCS data set, while the transformer achieved the best performance, the performance difference between it and the improved model is within one percentage point, showing excellent results. This difference indicates that, although the improved model excels in more complex fault scenarios, it also adapts effectively to simpler fault situations, highlighting its robustness and stability, and further demonstrating the model's strong generalization capability. Through discussion and analysis, the data expansion method based on auto-regressive generative adversarial network is used to conduct controlled experiments under different sample sizes, and the advantages of GATCN-Transformer in handling small sample fault data are verified. The conclusions are as follows.

(1) Under complex fault conditions, Graph Attention Convolutional Neural Network can effectively extract the spatial correlation between different parameters, and two-stage learning combined with the Transformer model can effectively improve the model's diagnostic performance in small samples and complex fault modes.

(2) The model in this paper can achieve higher diagnostic accuracy in different datasets, because it can better learn the spatial features between fault parameters in different datasets. Compared with the comparison models, the combination of temporal features and spatial features can better match the fault mode.

(3) After transforming the original data through the time window, our fault classification method is more inclined to pattern matching, and compared to traditional fault classification models, our fault classification method is more efficient and has a lower error detection rate.

(4) Graph Neural Network requires greater memory and time consumption during model training and the construction of parameter space features. In the future, we consider introducing transfer learning to speed up model training.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61873142).

References

1. Yaguo Lin, Bin Yang, Xinwei Jiang, et al. "Applications of machine learning to machine fault diagnosis: A review and roadmap." Mechanical Systems and Signal Processing 138 (2020): 106587.
2. Gonzalez-Jimenez David, Jon Del-Olmo, Javier Poza, et al. "Data-driven fault diagnosis for electric drives: A review." Sensors 21(12): 4024.
3. Tianci Zhang, Jinglong Chen, Fudong Li, et al. "Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions." ISA Transactions 119 (2022): 152-171.
4. Velickovic Petar, Guillem Cucurull, Arantxa Casanova, et al. "Graph attention networks." Stat 1050.20 (2017): 10-48550.
5. Vaswani Ashish, Noam Shazeer, Niki Parmar, et al "Attention Is All You Need.(Nips), 2017." arxiv preprint arxiv:1706.03762 10 (2017): S0140525X16001837.
6. Jiang Cui, Fan Zhou, Yongfan Chen, et al. "A fault diagnosis technique for aviation generator rectifiers based on GAMF-CNN." Chinese Journal of Aeronautics 1-11.
7. Da Zhang, Junyu Gao, Teng Ding, et al. "Fault detection of aviation sensors based on temporal two-dimensional transformation." Journal of Northwestern Polytechnical University 41.06(2023):1033-1043.
8. Huahui Yang, and Cheng Wang. "Data-driven feature extraction for analog circuit fault diagnosis using 1-D convolutional neural network." IEEE Access 8 (2020): 18305-18315.
9. Shengkai Zong. "Research on fault diagnosis of aircraft electromechanical systems based on multi-classifier fusion." MS thesis. University of Electronic Science and Technology of China, 2021.
10. Wenxiu Fu, Hongyang Li, DongMing Jin. "Fault diagnosis of train speed and distance measurement equipment based on LSTM." Journal of Beijing Jiaotong University 44.2 (2020): 9-16.
11. Siyu Shao, Ruqiang Yan, Yadong Lu, et al. "DCNN-based multi-signal induction motor fault diagnosis." IEEE Transactions on Instrumentation and Measurement 69.6 (2019): 2658-2669.

12. Yao, Zhiqiang. "Research on pipeline leak fault diagnosis method based on deep belief network." Journal of Safety Science and Technology 14.4(2018):6.
13. Min Xia, Xi Zheng, Muhammad Imran, et al. "Data-driven prognosis method using hybrid deep recurrent neural network." Applied Soft Computing 93 (2020): 106351.
14. Yibing Li, Dinghong Huang, Jiangbo Ma, et al. "Gear fault diagnosis method based on deep belief network and Information fusion." Journal of Vibration and Shock 40.8 (2021): 62.
15. Zhiqin Zhu, Yangbo Lei, Guangqiu Qi, et al. "A review of the application of deep learning in intelligent fault diagnosis of rotating machinery." Measurement 206 (2023): 112346.
16. Jie Zhou, Ganqu Cui, Shengding Hu, et al. "Graph neural networks: A review of methods and applications." AI Open 1 (2020): 57-81.
17. Chenyang Li, Lingfei Mo, and Ruqiang Yan. "Rotating machinery fault diagnosis based on spatial-temporal GCN." 2021 International Conference on Sensing, Measurement & Data Analytics in the Era of Artificial Intelligence (ICSMD). IEEE, 2021.
18. Yaping Wang, Sheng Zhang, Ruofan Cao, et al. "A Rolling Bearing Fault Diagnosis Method Based on the WOA-VMD and the GAT." Entropy 25.6 (2023): 889.
19. Shengmao Lin, Shu Wang, Xuefang Xu, et al. "GAOformer: An adaptive spatiotemporal feature fusion transformer utilizing GAT and optimizable graph matrixes for offshore wind speed prediction." Energy 292 (2024): 130404.
20. Defu Cao, Yujing Wang, Juanyong Duan, et al. "Spectral temporal graph neural network for multivariate time-series forecasting." Advances in neural information processing systems 33 (2020): 17766-17778.
21. Ke Zhang, Fang He, Zhengchao Zhang, et al. "Graph attention temporal convolutional network for traffic speed forecasting on road networks." Transportmetrica B: Transport Dynamics 9.1 (2021): 153-171.
22. Jiang Li, Shuaiyu Wang, Tianao Zhang, et al. "Semi-supervised few-shot fault diagnosis driven by multi-head dynamic graph attention network under speed fluctuations." Digital Signal Processing 151 (2024): 104528.
23. Ge Guo, and Wei Yuan. "Short-term traffic speed forecasting based on graph attention temporal convolutional networks." Neurocomputing 410 (2020): 387-393.
24. Seongjun Yun, Minbyul Jeong, Raehyun Kim, et al. "Graph transformer networks." Advances in neural information processing systems 32 (2019).
25. Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, et al. "Recurrent neural networks for multivariate time series with missing values." Scientific reports 8.1 (2018): 6085.
26. Yinjun Wu, Jingchao Ni, Wei Cheng, et al. "Dynamic gaussian mixture based deep generative model for robust forecasting on sparse multivariate time series." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 1. 2021.
27. Zonghan Wu, Shirui Pan, Guodong Long, et al. "Connecting the dots: Multivariate time series forecasting with graph neural networks." Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020.
28. Jinsung Jeon, Jeonghak Kim, Haryong Song, et al. "GT-GAN: General purpose time series synthesis with generative adversarial networks." Advances in Neural Information Processing Systems 35 (2022): 36999-37010.