

RESEARCH

Open Access



TN-GTN: fault diagnosis of aircraft wiring network over edge computing

Tian Wang¹, Qiang Fang¹, Gongping Liu², Meng Chi^{3*} , Yuanqi Luo² and Jianming Shen⁴

*Correspondence:
chimeng089@gmail.com

¹ School of Mechanical Engineering, Zhejiang University, 38 Zheda Road, Hangzhou 310027, Zhejiang, China

² AVIC Xi'an Aircraft Industry Group Company Ltd, No.1 Xifei Avenue, Xi'an 710089, Shaanxi, China

³ College of Computer Science and Technology, Zhejiang University, 38 Zheda Road, Hangzhou 310027, Zhejiang, China

⁴ Company of AVIC Shenyang Aircraft Corporation, 1 Lingbei Road, Shenyang 110034, Liaoning, China

Abstract

Fault diagnosis of the aircraft wiring network plays an important role in the intelligent manufacture of the aircraft. Many studies focus on the feature-based machine learning methods. However, these methods are improper in handling the data on heterogeneous graphs. Due to the scatter of the valid feature information, the relevant information between the test nodes is ignored by these methods, which leads to the low accuracy fault diagnosis. Taking the advantage of the 5G technology that can remotely process large-scale graph data, this work proposes a fault diagnosis method named “topological network-graph transformer network (TN-GTN).” TN-GTN can improve the fault diagnosis accuracy through feature enhancement and classification, which is based on the topological information of heterogeneous graphs. The graph network is able to learn new graph structures by identifying useful meta-paths and multi-hop connections between unconnected nodes on original graphs. Feature-enhanced test nodes are used to classify the final labels by the artificial neural network. Results of the performed experiment showed that TN-GTN reduced the dependence on domain knowledge and achieved an accurate classification of the fault diagnosis on aircraft wiring network.

Keywords: Edge computing, Fault diagnosis, Aircraft wiring network, Feature enhancement, Graph transformer network (GTN)

1 Introduction

In recent decades, with the development of computer technology and electronic control technology, more and more electronic products replace the traditional mechanical structure on the aircraft, which doubles the number of on-board electronic equipment, making the wiring network of the aircraft more and more complex. Accordingly, the diagnosis data increase, which requires edge computers to process large-scale data. The emergence of edge computing provides a new solution to large-scale data processing in the Internet of things (IoT) environment [1–7]. Meanwhile, algorithm researches on processing large-scale data with edge devices become possible. The improvement of fault diagnosis technology based on edge computing also guarantees the safety of the aircraft wiring network.

Since the 1960s, the fault diagnosis technology of aircraft wiring has changed from manual detection to automatic testing in the aircraft assembly stage. Many fault diagnosis methods of cable wiring are widely used, including time domain reflectometry (TDR)

[8], frequency domain reflectometry (FDR) [9], sequence time domain reflectometry (STDR) [10] and noise domain reflectometry (NDR) [11]. Through this work, cold welding joints, worn points and other anomalies are detected with time or frequency domain reflection methods or phase diagnosis frequency antireflection method. The study of the cable fault positioning through the signal analysis gains great achievements. Broken points or worn points can be located on the single wiring.

Due to the large number of test nodes and the complex topological network of the test equipment, the valid feature information in the fault diagnosis of aircraft wiring network is scattered, and the relevant information between the test nodes is ignored, which leads to the low accuracy fault diagnosis. The existing automatic test system of the aircraft wiring network can report the fault but cannot define the accurate fault type. Objectively, the wiring failure can be caused by the open circuit or short circuit of the wiring on the aircraft, and also by pseudo-faults, such as unreliable connections to the test equipment and the damage in the test equipment itself. Because of its electrical and mechanical properties and the chemical stability, aircraft wiring can run stably at temperatures ranging from -65 to $+200$ degrees Celsius. Ignoring the serious impact of aging, a single aircraft wiring is rarely worn or broken [12]. Due to continuous improvement of aviation cable physical quality and the mature assembly process, the aircraft wiring network faults, especially the faults on newly produced aircraft, are mostly pseudo-faults. Before analyzing the causes of the faults, pseudo-faults must be first eliminated, which usually takes half the total test time or more. Therefore, an effective method is needed to classify pseudo-faults and real faults. The fault classification cannot be completed with time domain reflection method or frequency domain reflection method. It requires new fault diagnosis approaches to classifying data.

2 Related work

Kuželka and Davis [13] studied at the theoretical level the applicability of learning Markov logical network weights in the knowledge base on the condition of lacking data. After weight learning, the first-order logical rules are mapped on the Markov network which uses the soft logic as its logical component and the Markov network as its statistical model. The method can simulate human reasoning and is also able to assist reasoning with domain knowledge. To ensure the accuracy of reasoning, it needs to define and analyze each type of the faults, but the rules of reasoning are not easy to make because of its reliance on large scale of domain knowledge. There is another method to analyze and classify complex data based on data mining [14] and naive Bayes method [15]. This method needs an enormous rule formulation basis. Indeed, such methods are difficult to apply.

As research goes future, attention has been focused on graph data embedding and classification as a new research direction. Researches based on graph structure data, including node classification [16], link prediction [17] and graph classification [18], come out. In recent years, because of the powerful expression of graph structure, the study of graph analysis with machine learning method has been paid more and more attention. Graph neural network (GNN) has recently become a widely used graph analysis method due to its superior performance and interpretability [19]. Other

representative methods include graph convolutional network (GCN) [20], Graph SAGE [21], graph attention network (GAT) [22, 23] and their variants [24].

However, these GNN-based methods can be only applied to homogeneous graphs [25], whose node types and edge types are unique. In the manufacture process, the graph data often contain multiple node types and edge types. Furthermore, each type of nodes may be associated with attributes in different feature spaces. Such graphs with heterogeneity can also be called heterogeneous information networks (HINs) [26]. To tackle the heterogeneity of graphs, a common method is to manually design meta-path [27], a composite relation scheme, widely used to represent node relationships in heterogeneous graphs [28]. Based on HAN [29], Metapath2vec [30] uses the manually defined meta-path to obtain graph representation and structure learning. Obviously, it requires enormous labor to generate meta-path, and its application scope is limited because of its dependence on work experience and domain knowledge. Then, a graph transformer network (GTN) is adopted to model the graph to obtain node embedding [31]. GTN learns to transform a heterogeneous input graph into useful meta-path graph for each task and learns node representation on the graphs. GTN can also aggregate the representations of meaningful neighbors of nodes by multi-channel mechanism.

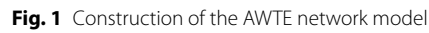
In this paper, the topological network-graph transformer network (TN-GTN) is proposed to make fault diagnosis of aircraft wiring network by data classification. TN-GTN can generate new meta-path based on the network topology information of test equipment. TN-GTN can also aggregate representations of meaningful neighbor nodes on heterogeneous graph. The trained GTN will add new feature information to test nodes, which is named the feature-enhanced test nodes. TN-GTN uses feature-enhanced test nodes as new input and classifies the final labels by artificial neural network.

The contents of this paper are summarized as follows:

- (1) AWTE (Aircraft Wiring Test Equipment) network is established to analyze associations between test nodes. In the network, the nodes include test nodes, plugs on aircraft wiring network and test equipment, while the edges represent the associations between nodes.
- (2) TN-GTN is employed to learn new graph structures on the basis of predefined meta-path. New graph structures involve useful meta-paths and multi-hop connections between unconnected nodes on the original graph. TN-GTN can be used to aggregate the information of different nodes and obtain the feature-enhanced test nodes.
- (3) Evaluation results show that TN-GTN outperforms ANN approach in fault diagnosis of aircraft wiring network.

3 TN-GTN method

In this section, basic definitions and concepts of TN-GTN method are introduced. In addition, algorithm application background is also illustrated by constructing AWTE network. Meta-path generation, heterogeneous convolution and final fault classification through TN-GTN are described with details.



AWTE network model is constructed to analyze associations between test nodes, which involves a lot of preparatory work, including original data collection, establishing the test equipment network schema model and constructing heterogeneous graph (Fig. 1).

According to the function of test equipment system structure, test process and wiring network fault diagnosis, the schema model can be designed (Fig. 2).

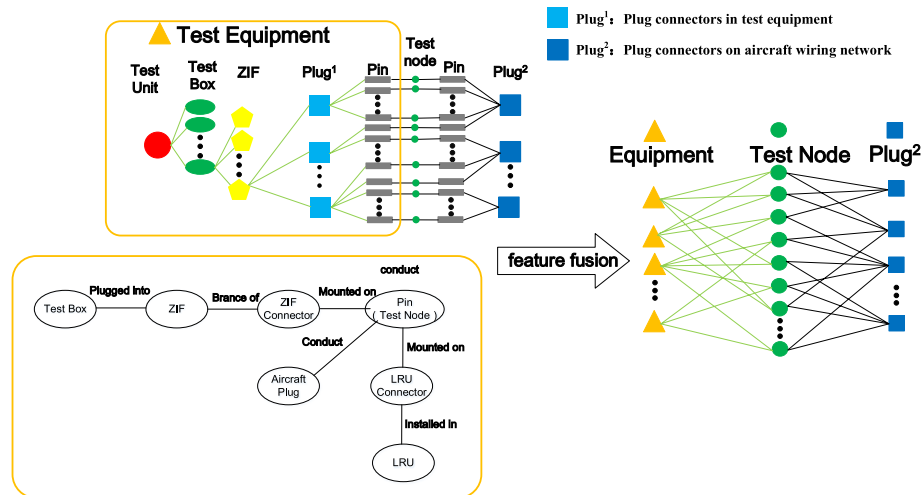
The input graph is regarded as a heterogeneous graph $G=(V, E)$, where V is a set of nodes, E is a set of observed edges with a node type mapping function $f_v: V \rightarrow T^v$ and an edge-type mapping function $f_e: E \rightarrow T^e$. Each node $V_i \in V$ has a node type, $f_v(V_i) \in T^v$.

Table 1 Entities and properties of AWTE

Entity	Description
Test box	Primary child node in test equipment
LRU	Primary child node in test equipment: linear replaceable unit
ZIF	Secondary child node in test equipment: zero insertion force modular attached on test box
ZIF connector	Third-level child node: connector attached on zero insertion force modular
LRU connector	Secondary child node in test equipment: connector attached on linear replaceable unit
Aircraft plug	Tested object installed on aircraft
Pin (Test node)	Test information collection point

Table 2 Relations between AWTE entities

Relation	Description
Plugged into	Test box-ZIF, indicating that the ZIF is installed on the Test Box
Branch of	ZIF connector-ZIF, indicating that the ZIF connector is one of many branches of ZIF
Mounted on	Pin-connector, indicating that the pinhole is connected to ZIF connector/LRU connector
Conduct	Pin-Pin, indicating the conduction relationship between pins

**Fig. 3** Node simplification on schema model of AWTE

Similarly, $e_{ij} \in E, f_e(e_{ij}) \in T^e$. When $|T^e|=1$ and $|T^v|=1$, it becomes a homogeneous graph. In this paper, the set condition is $|T^e|>2$ and $|T^v|>2$.

According to AWTE schema model, all nodes are simplified into three categories, (1) Equipment-test devices on AWTE, (2) Plug–Plug² connectors on aircraft wiring network and (3) Test Node-Junction between Plug¹ connectors on test equipment and Plug² connectors on aircraft wiring network.

The test equipment is a multi-level tree structure network, with many Plugs¹ at the end. Each Plug¹ contains several pins, and the test equipment forms a complete loop by docking the pins on the test equipment end with the pins on the aircraft Plug² at the test node (Fig. 3).

The heterogeneous graph can be represented by a set of adjacency matrices $\{A_K\}_{K=1}^K$ where $K=|T^e|$. $A_K \in R^{n \times n}$ is an adjacency matrix where $A_K[i, j]$ is nonzero when there is a k -th type edge from j to i . More compactly, it can be written as a tensor $A_K \in R^{n \times n \times K}$. A feature matrix $X \in R^{n \times d}$ means that the d -dimensional input feature is given to each node.

The AWTE heterogeneous graph describes a graph that contains four kinds of edges: $T^e = \{NE, EN, PN, NP\}$.

The yellow adjacency matrix A_{NE} represents the predefined meta-path $N \xrightarrow{NE} E$, which is used to describe the connection between test nodes and test equipment. The red adjacency matrix A_{EN} represents the predefined meta-path $E \xrightarrow{EN} N$, which is used to describe the connection between test equipment and test nodes. The green adjacency matrix A_{PN} represents the predefined meta-path $P \xrightarrow{PN} N$, which is used to describe the connection between plugs and test nodes. The blue adjacency matrix A_{NP} represents the predefined meta-path $N \xrightarrow{NP} P$ which is used to describe the connection between test nodes and plugs. When there is an edge of type k from j to i , the element on $A_K[i, j]$ will not be zero. For example, if there is an edge between the test equipment with serial number 1 and the test node with serial number 1 in A_{NE} , E_1N_1 will not be zero. Otherwise, E_1N_1 is zero (Fig. 4).

E_iN_j indicates whether there is a connection between the i th test equipment and the j th test node. P_mN_j indicates whether there is a connection between the m th aircraft plug connector and the j th test node

$$\begin{bmatrix} E_1N_1 & E_2N_1 & E_3N_1 & \dots \\ E_1N_2 & E_2N_2 & E_3N_2 & \dots \\ E_1N_3 & E_2N_3 & E_3N_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} P_1N_1 & P_2N_1 & P_3N_1 & \dots \\ P_1N_2 & P_2N_2 & P_3N_2 & \dots \\ P_1N_3 & P_2N_3 & P_3N_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 0 & \dots \\ 1 & 0 & 1 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (1)$$

3.2 Classification of key nodes with enhanced feature

Classification based on data-driven approach requires valid feature information as much as possible, so it is necessary to enhance features of classified objects. An effective way is to collect effective information from relevant nodes. The graph neural network is efficient to update the new embedding of the nodes by aggregating information on the graph. GTN can transform the heterogeneous graph into a new graph structure through meta-path which contains information about nodes and edges. The information about

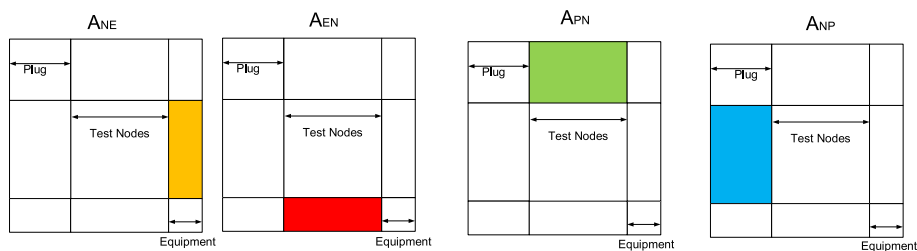


Fig. 4 Adjacency matrices of AWTE heterogeneous graph

center edges can aggregate feature information of nodes by different paths, using meta-path to enhance feature on the basis of relevant nodes. Useful test paths include Node–Plug–Test Node (NPN) and Test Node–Plug–Equipment (NPE). Therefore, a new model is needed to learn node representations and graph structures on heterogeneous graph by means of convolution. The model can generate a new soft-selected edge-type graph structure (Fig. 5).

At the data collection stage, test nodes only contain the originally collected feature information such as resistance measurement value and insulation characteristic test result. After enhancing the feature, the TN-GTN model adds a dimensional feature to the test nodes, which is used to characterize strength of the correlation between test nodes.

3.2.1 AWTE feature enhancement

From the point of original data collection, test nodes are independent to each other. But test nodes are merged into one huge heterogeneous graph in the sense of AWTE topological network and aircraft wiring network. The correlation between test nodes can be obtained by GTN in a way of learning the location information, connector type information and test equipment topology network information.

The associations between test nodes can be defined as strong association, weak association and non-association:

Article (1): If test nodes are mounted on the same type of connectors or even the same connector, and these test nodes meet the principle of proximity, the association between test nodes is defined as strong association.

Article (2): If test nodes meet the requirements of either principle in the previous article (1), the association between test nodes is defined as weak association.

Article (3): If test nodes meet neither of the principles in the previous article (1), the association between test nodes is defined as non-association.

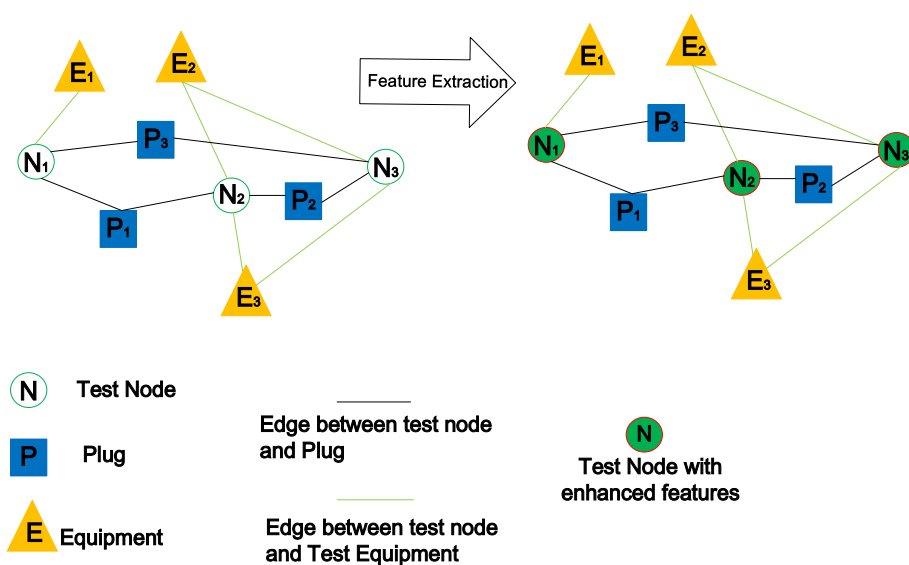


Fig. 5 Feature enhancement of key nodes

Useful meta-paths are relevant to predefined meta-paths between target nodes (nodes with classifying labels). TN-GTN can discover new graph structures and new relevant meta-paths between all types of nodes.

TN-GTN judges whether test nodes N_i and N_j belong to the same type of connectors or even the same connector and whether they meet the principle of proximity through valid meta-paths such as NEN, NPN and NPEPN.

$$f(N_i, N_j) = \begin{cases} 1 & (N_i, N_j)_{\text{article}(1)} \\ 0.5 & (N_i, N_j)_{\text{article}(2)} \\ 0 & (N_i, N_j)_{\text{article}(3)} \end{cases} \quad (2)$$

N_i and N_j represent different test nodes. Scores of $f(N_i, N_j)$ are different according to different strength of the correlation between nodes. The score is an additional feature to test nodes after enhancing feature.

The core idea of TN-GTN is message passing between plugs with feature information of different connector types and test equipment with location information. TN-GTN is an effective way to aggregate useful information to enhance the feature of test nodes. According to the experience of troubleshooting, test nodes with strong association easily lead to some specific types of faults.

3.2.2 Meta-path on AWTE

Meta-path [32] denoted by P is a path on the heterogeneous graph G which is connected with heterogeneous edges, i.e., $V_1 \xrightarrow{t_1} V_2 \xrightarrow{t_2} \dots V_k \xrightarrow{t_k} V_{k+1} \dots \xrightarrow{t_l} V_{l+1}$ where $t_k \in T^e$ denotes an k -th edge type of meta-path. Meta-path defines a composite relation $R = t_1 o t_2 \dots o t_l$ between node V_1 and V_{l+1} , where $R_1 o R_2$ denotes the composition of relation R_1 and R_2 . Given the composite relation R or the sequence of edge types (t_1, t_2, \dots, t_l) , the adjacency matrix A_P of the meta-path P is obtained by the multiplications of adjacency matrices as

$$A_P = A_{t_l} \dots A_{t_2} A_{t_1} \quad (3)$$

The notion of meta-path subsumes multi-hop connections and new graph structures in AWTE framework are represented by adjacency matrices, i.e., the meta-path: Test Node–Plug–Equipment (NPE), which can be represented as $N \xrightarrow{NP} P \xrightarrow{PE} E$, generates a new adjacency matrix A_{NPE} by the multiplication of A_{NP} and A_{PE} .

3.2.3 Meta-path generation and heterogeneous convolution

The Metapath2vec approach utilizes a manually defined meta-path [30]. Path learning based on HAN [29] gets graph representation. However, it takes enormous labor to generate a meta-path.

TN-GTN is used to generate different homogeneous subgraphs from original heterogeneous graphs and uses *softmax* to randomly select subgraphs Q_1 and Q_2 in Fig. 6. The matrix multiplication of Q_1 and Q_2 is used to generate a new meta-path. Graph Transformer Layer (GTL) is mainly used to randomly generate multiple meta-paths

The generation of new meta-path graph in Graph Transformer (GT) Layer in Fig. 6 contains two components. First, GT layer softly selects two graph structures Q_1 and Q_2 from candidate adjacency matrices A . Second, it learns a new graph structure by the

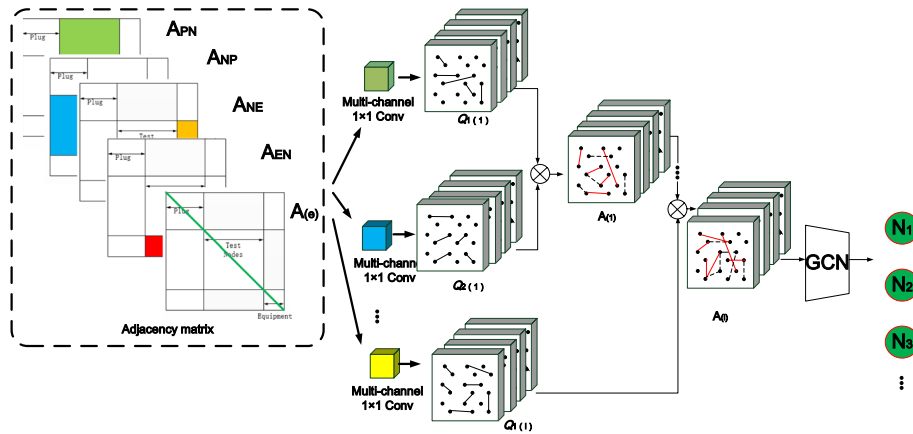


Fig. 6 GTN learn to generate a set of new meta-path adjacency matrices

composition of two relations. $A_{(1)}$ is a matrix multiplication of two adjacency matrices Q_1 and Q_2 .

The AWTE heterogeneous graph describes a graph that contains four kinds of edges: $T^e = \{NE, EN, PN, NP\}$. The corresponding adjacency matrix is $A = \{A_1, A_2, A_3, A_4\}$. The soft adjacency matrix selection is a weighted sum of candidate adjacency matrices obtained by multi-channel 1×1 convolution with nonnegative weights from *softmax* ($W_\varphi^1, W_\varphi^2, \dots$), where $W_\varphi^1 = \{\alpha_1^1, \alpha_2^1, \alpha_3^1, \alpha_4^1\}$, $W_\varphi^2 = \{\alpha_1^2, \alpha_2^2, \alpha_3^2, \alpha_4^2\}$.

$$Q_1 = \alpha_1^1 * A_1 + \alpha_2^1 * A_2 + \alpha_3^1 * A_3 + \alpha_4^1 * A_4 \quad (4)$$

$$Q_2 = \alpha_1^2 * A_1 + \alpha_2^2 * A_2 + \alpha_3^2 * A_3 + \alpha_4^2 * A_4 \quad (5)$$

$$A_{(1)} = Q_1 * Q_2 \quad (6)$$

The new meta-path graph computes the convex combination of adjacency matrices as $\sum_{t_l \in T^e} \alpha_{t_l}^{(l)} A_{t_l}$, where T^e denotes a set of edge types, $\alpha_{t_l}^{(l)}$ is an attention score for edge type t_l at the l th GT layer, and A_{t_l} denotes adjacency matrix of layer l .

TN-GTN can learn an arbitrary meta-path with respect to edge types and path length. The adjacency matrix of arbitrary length l meta-paths can be calculated by Eq. 7 (Fig. 6).

$$A_P = \left(\sum_{t_1 \in T^e} \alpha_{t_1}^{(1)} A_{t_1} \right) \left(\sum_{t_2 \in T^e} \alpha_{t_2}^{(2)} A_{t_2} \right) \cdots \left(\sum_{t_l \in T^e} \alpha_{t_l}^{(l)} A_{t_l} \right) \quad (7)$$

Graph transformer networks (GTNs) learn to generate a set of new meta-path adjacency matrices $A_{(l)}$ by GT layers and perform graph convolution as in GCNs on new graph structures.

Firstly, different datasets are constructed through multi-channel sampling. Secondly, different meta-paths are generated with GT layers, and the output features are spliced as the input of a graph convolutional network (GCN) [33]. Finally, GCN is used to extract the representation of graph end to end and to learn useful representations for node classification in an end-to-end fashion.

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

$\tilde{A} = A + I \in R^{N \times N}$ is the adjacency matrix A of the graph G with added self-connections, \tilde{D} is the degree matrix of A , and $W^{(l)}$ is a trainable weight matrix. GCN can easily observe that the convolution operation across the graph is determined by the given graph structure and it is not learnable except for the node-wise linear transform $H^{(l)} W^{(l)}$. The convolution layer can be interpreted as the composition of a fixed convolution followed by an activation function σ on the graph after a node-wise linear transformation. $H^{(l+1)}$ is the feature representations of the $(l+1)$ th layer in GCN, and the forward propagation becomes Eq. 8.

3.2.4 Fault classification of aircraft wiring network based on TN-GTN

After enhancing features of AWTE network by using the GTN, the ANN model is used as a supervised learning model for data classification. The input graph is regarded as a heterogeneous graph $G = (V, E)$, and E is four kinds of edges: $T^e = \{NE, EN, PN, NP\}$. V contains three types of nodes: $T^v = \{X^E, X^N, X^P\}$. All V_i in $X \in R^{n \times d}$ means that the d -dimensional input feature is given to each node. TN-GTN generates new meta-path on the basis of predefined meta-path. Finally, the m -dimensional feature-enhanced test nodes $X^N \in R^{n \times (d+m)}$ are used as new inputs into ANN to complete the task of fault classification. The number of input layer nodes is n , which is also the number of test nodes. The number of nodes in output layer is 5, which represents five final classifications, including “test result Success, Fault1, Fault2, Fault3 and Fault4” (Fig. 7).

Common faults were divided into four types according to work experience as follows:

Fault1 Wrong connection between the test equipment and test nodes

The ZIF connector A1 is wrongly connected or not connected to the test nodes, causing the continuity test failure. When this failure occurs, the continuity test of all test nodes on ZIF connector fails. This type of fault often occurs on the condition of strong correlation between test nodes. The fault type is influenced by the resistance value of the test node and the secondary continuity test result (Fig. 8).

Fault2 Wrong connection to a similar connector

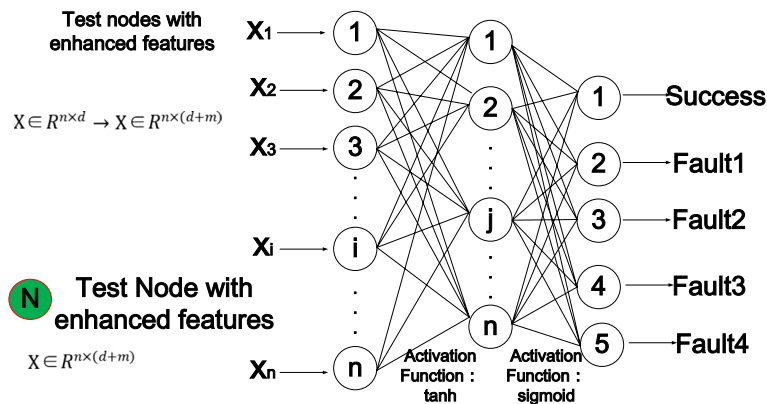


Fig. 7 Classification on test nodes with enhanced information

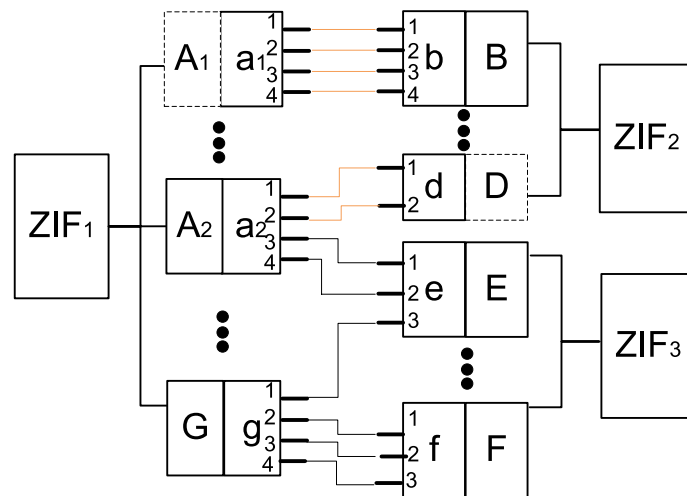


Fig. 8 Wrong connection between the test equipment and test nodes

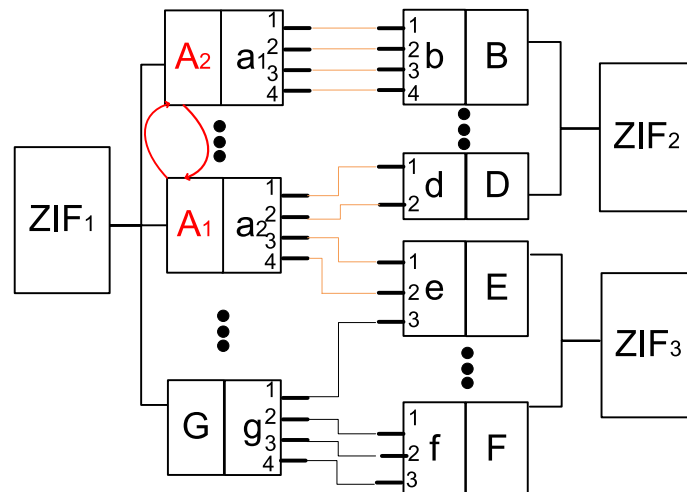


Fig. 9 Wrong connection to a similar connector

There are many similar but differently defined connectors on the aircraft, such as A1 and A2. The failure continuity test results occur when AWTE connector wrongly matches the aircraft plug connector. When this failure occurs, the continuity test of all test nodes on the same type of connector fails. This type of fault often occurs on the condition of strong correlation between test nodes. The fault type is not influenced by the resistance value of the test node and the secondary continuity test result (Fig. 9).

Fault3 Wrong connections among the aircraft plug connectors.

Adjacent pins in the same connector are misplaced. When this failure occurs, the adjacent test nodes on the connector cannot pass the continuity test. This type of fault often occurs on the condition of strong correlation between test nodes. The fault type is influenced by the resistance value of the test node and the secondary continuity test result. Test nodes must have poor insulation performance (Fig. 10).

Fault4 Wrong connection in the aircraft wiring network

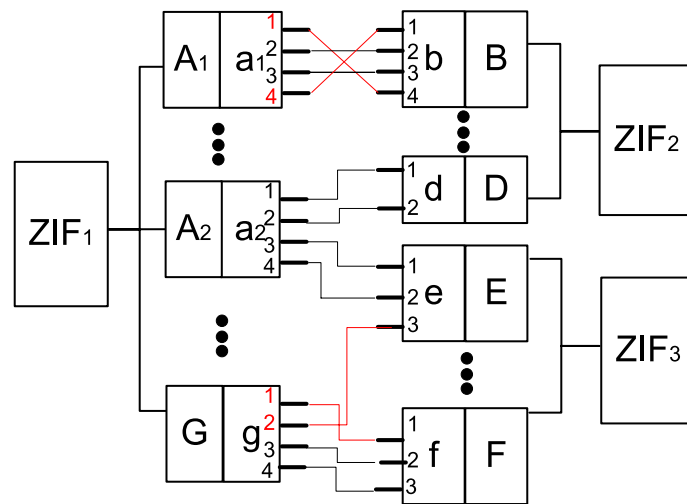


Fig. 10 Wrong connections among the aircraft plug connectors

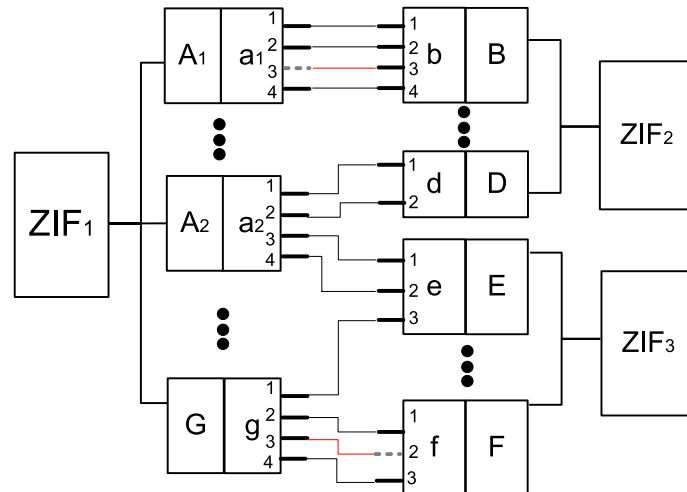


Fig. 11 Wrong connection in the aircraft wiring network

When broken points or wear is on a single wire, other test nodes in wiring network will not be affected. This type of fault often occurs on the condition of weak correlation or non-correlation between test nodes. The fault type is only influenced by the resistance value of the test node and the secondary continuity test result (Fig. 11).

4 Experimental results and discussion

In this section, large-scale experiments are designed to evaluate TN-GTN's effectiveness of fault diagnosis on aircraft wiring network.

The task to diagnose aircraft wiring network is to classify the fault types and achieve accurate fault diagnosis according to the feature of test nodes. This paper lists four fault types.

Precision, recall and F1-score are used to evaluate the classifying efficiency of fault diagnosis. Finally, an experiment is designed to compare the classifying results of

TN-GTN and ANN algorithm. The TN-GTN is implemented in PyCharm and executed under the environment of Intel(R) Core (TM) i7-10750H CPU @ 2.60 GHz, 16 GB RAM and Windows 10, 64bit operating system.

4.1 Dataset and feature extraction

Heterogeneous graph datasets with multiple types of nodes and edges were used to evaluate the efficiency of TN-GTN. Dataset contained three types of nodes (Test Node (N), Plug (P) and Equipment (E)), four types of edges (NP, PN, NE and EN) and five types of test results (Success and Fault1, Fault2, Fault3 and Fault4) as labels. The main task was to classify nodes.

The feature data of the Test Node (N) were obtained from the actual test. A complete loop was formed by aircraft wiring network and test equipment. 13892 piece of test results were obtained at one time through a 2 h continuity test and a 5-h insulation test. After the test, a check on all test nodes was carried out to confirm the results. There was Success (11913), Fault1 (1285), Fault2 (224), Fault3 (347) and Fault4 (123). Each piece of data included the resistance value obtained in continuity test, the insulation characteristics of test node and the results of secondary continuity test. This type of experiment was performed for five times.

The feature data of the Plug (P) was objective, which was not generated during the testing process. There were a total of 2016 plugs on the aircraft wiring network. The feature data of the Plug included the type of the plug and the three-dimensional coordinate information under Cartesian coordinates. The data were given as one-hot encoding representations of plots.

The feature data of the Equipment (E) were objective, which was not generated during the testing process. There were a total of 234 nodes after simplification on AWTE network. The feature data of the Equipment (E) included the branch information, the type of test box and the manufacture information of test equipment. The data were given as one-hot encoding representations of plots (Table 3).

The topological network of three node types was connected by four types of predefined meta-path $N \xrightarrow{NE} E$, $E \xrightarrow{EN} N$, $P \xrightarrow{PN} N$ and $N \xrightarrow{NP} P$.

The trained GTN would add new feature information to test nodes, which was named the feature-enhanced test nodes. Enhanced feature is Test Node score of $f(N_i, N_j)$, 0 score means non-association, 0.5 score means weak association, and 1 score means strong association according to the different strength of the correlation between nodes. (Section. AWTE feature extraction—Eq. 2).

Useful meta-paths are relevant to predefined meta-paths between target nodes (nodes with classifying labels). TN-GTN can discover new graph structures and new relevant meta-paths between all types of nodes.

Table 3 Datasets on heterogeneous graph

Test node	Equipment	Plug	Edge	Nodes type
13892	234	2016	55568	3

In Table 4, predefined meta-paths and meta-path learnt by TN-GTN were summarized with domain knowledge. Predefined meta-paths were obtained on the basis of correlation between test equipment and aircraft wiring network.

4.2 Evaluation indicators

Precision, recall and F1-score were selected as evaluation indicators. TN-GTN has greatly improved the fault classification accuracy of aircraft wiring network. The evaluation indicators can visually demonstrate the TN-GTN's excellent performance of test nodes' classification on heterogeneous graph datasets. In the case of evaluating the classification performance for each class separately, the multi-classification task can be simplified into a classification task that distinguishes its own class from a non-self-class, which is transformed into a binary classification task. In this work, precision, recall and F1-score are widely used to evaluate the performance of classification algorithms, especially in the classification evaluation of imbalanced distribution datasets.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

$$F1 - \text{score} = \frac{2}{1/\text{precision} + 1/\text{recall}} \quad (11)$$

where TP is the number of cases where the predicted type of fault is positive and the actual type of fault is positive; TN: the number of cases where the predicted type of fault is negative and the actual type of fault is negative; FP: the number of cases where the predicted type of fault is positive and the actual type of fault is negative; and FN: the number of cases where the predicted type of fault is negative and the actual type of fault is positive. Precision refers to the proportion of the number of correctly predicted faults of a certain type in the predicted faults of a certain type, so it is also called precision rate; recall refers to the proportion of the number of correctly predicted faults of a certain type in the actual faults of a certain type; and F1-score is the harmonic average value, which is an evaluation index considering both precision and recall.

4.3 Discussion with evaluation of experimental results

The aircraft wiring network test dataset was divided into two parts: 20% of the 13,892 sampled dataset was used as the test data, and the remaining 80% was used as the training data. The training data were randomly divided into 4 groups to test ANN and

Table 4 Predefined meta-path and meta-path learnt by TN-GTN

Predefined meta-path	Meta-path learnt by TN-GTN (most useful)
NE, EN, PN, NP	NEN, ENE, NPN, PNP NEPEN, NPEPN, PENEP, PNEPN

TN-GTN under different training conditions (1) lack of training data (20% training: 1 group of the training data used), (2) scarce training data (40% training: 2 groups of the training data used), (3) normal training data (60% training: using 3 groups of the training data), (4) sufficient training data (80% training: 4 groups of the training data used).

Table 5 lists the precision, recall and F1-score of the four types of fault classification on different training conditions.

The ANN model contained a hidden layer. 100 nodes were built by using tanh function in the first layer and sigmoid in second layer as activation function. The number of nodes in output layer was 5, which represented five final classifications, including “test result Success, Fault1, Fault2, Fault3 and Fault4.” The learning rate of ANN model was set 0.4. The end of TN-GTN was also an ANN model with the same hidden layer parameters for comparison experiments.

The original data, such as resistance value of test nodes, continuity test results and leakage test results, were used for training ANN model. Graph data contained more information like topological network information, location, relevant information between test nodes and connector types. The graph data were used for training TN-GTN model.

When the training set was scarce, it could be seen that TN-GTN improved precision, recall and F1-score in all types of fault classification compared to ANN. In particular, Fault2, Fault3 and Fault4, which relied on the enhanced features obtained from TN-GTN to judge, had been significantly improved, and the precision improvement was above 0.1. Classification performance in Fault1 was relatively good, and Precision could reach above 0.9. The features of Fault1 were more obvious, and high precision of classification could be obtained with enough training data. It could be seen from the experiments that TN-GTN used more relevant feature information than ANN. It was rarely misdiagnosed the current type of fault with another type of fault, so the recall score was also relatively high. With the increase in training data, the precision of ANN for Fault1 and Fault3 had reached more than 0.9, the precision of Fault2 was basically unchanged, and the classification effect of TN-GTN for Fault2 had been significantly improved. ANN’s judgment of Fault4 depended closely on a large number of training data. When the number of samples was sufficient (80% training), the accuracy reached 0.8. TN-GTN was more accurate for Fault4 classification regardless the sample sufficiency (Fig. 12).

It could be seen from the experiment results that both ANN and TN-GTN had a good effect on the classification of Fault1 with the increase in training data. Recall score of TN-GTN in Fault1 had been relatively high, which showed that TN-GTN could effectively reduce the misjudgment of other types of faults as Fault1. The performance of TN-GTN on Fault2 was much better than ANN; especially when the training data were insufficient, TN-GTN could improve the classification accuracy by aggregating the information of surrounding test nodes. Wrong connections among the aircraft plug connectors were more likely to appear in the test nodes with strong association. TN-GTN aggregated other types of node information in heterogeneous graphs, which was more sensitive to identify such faults. Compared with pseudo-faults such as Fault1 and Fault2 appearing on the AWTE side, real faults such as Fault3 and Fault4 appearing on the aircraft wiring network side were key points in troubleshooting. TN-GTN could transform the topological network information on the aircraft wiring network side through

Table 5 Experiment results in the fault classification task

	Fault1				Fault2				Fault3				Fault4			
		Precision	Recall	F1-score		Precision	Recall	F1-score		Precision	Recall	F1-score		Precision	Recall	F1-score
20% training	ANN	0.92	0.76	0.83	0.61	0.77	0.74	0.67	0.79	0.73	0.76	0.62	0.62	0.75	0.74	0.62
	TN-GTN	0.96	0.85	0.91	0.77	0.87	0.87	0.82	0.9	0.85	0.88	0.76	0.62	0.75	0.74	0.76
40% training	ANN	0.95	0.78	0.86	0.6	0.78	0.78	0.68	0.89	0.78	0.83	0.67	0.67	0.67	0.75	0.73
	TN-GTN	0.96	0.88	0.92	0.77	0.87	0.87	0.82	0.95	0.9	0.92	0.73	0.73	0.67	0.75	0.77
60% training	ANN	0.97	0.82	0.89	0.62	0.78	0.78	0.69	0.94	0.8	0.87	0.67	0.67	0.67	0.75	0.71
	TN-GTN	0.99	0.91	0.95	0.84	0.91	0.91	0.87	0.95	0.9	0.92	0.86	0.86	0.86	0.88	0.89
80% training	ANN	0.99	0.82	0.9	0.61	0.78	0.78	0.76	0.93	0.98	0.95	0.8	0.8	0.8	0.8	0.8
	TN-GTN	0.99	0.94	0.96	0.89	0.91	0.91	0.88	0.98	0.96	0.95	0.95	0.9	0.9	0.91	0.91

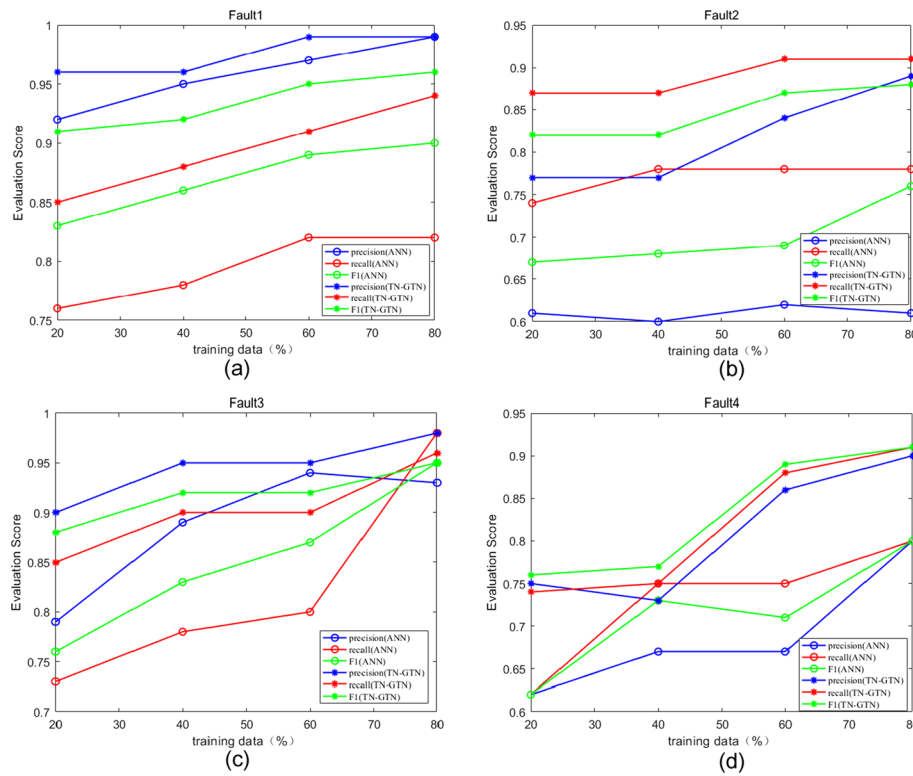


Fig. 12 Experimental results for different fault classification: **a** Fault1; **b** Fault2; **c** Fault3; and **d** Fault4

meta-paths, which enabled the TN-GTN classification method of performing well even with only a small amount of training data.

5 Conclusion

In this paper, TN-GTN (topological network-graph transformer network) is proposed to learn node representations on a heterogeneous graph of aircraft wiring network and Aircraft Wiring Test Equipment network. This approach is employed to learn new graph structures on AWTE, which involves identifying useful meta-paths and multi-hop connections between unconnected nodes on the original graph. The actual test data generated in the diagnosis process of aircraft wiring network can be used to prove efficiency of TN-GTN. The results show that TN-GTN is superior to traditional classification approach in terms of classification accuracy. TN-GTN can achieve high precision in some types of fault diagnosis, especially with fewer samples. At the same time, this data-driven approach reduces the dependence on domain knowledge. To integrate the feature information in time dimension into TN-GTN approach will be an interesting direction in the future. We will take more datasets from multiple aircraft into the consideration of heterogeneous graph to provide more proof for fault nodes predicting.

Abbreviations

GTN	Graph transformer network
TN-GTN	Topological network-graph transformer network
ANN	Artificial neural network
GNN	Graph neural network
HINs	Heterogeneous information networks

GCN	Graph convolutional network
AWTE	Aircraft Wiring Test Equipment
LRU	Linear replaceable unit
ZIF	Zero insertion force modular

Authors' information

Tian Wang received the bachelor's degree in the School of Mechanical and Automotive Engineering, South China University of Technology, in 2017. He is currently pursuing the Doctor's degree in School of Mechanical Engineering, Zhejiang University. His research interests include edge computing, deep learning and automatic diagnostic system for aircraft wiring.

Acknowledgements

Not applicable.

Author contributions

TW, GL and QF conceived and designed the study. TW, YL and QF performed the experiments. TW, JS and MC wrote the paper. QF reviewed and edited the manuscript. All authors read and approved the final manuscript.

Funding

This work is supported by the National Defense Basic Scientific Research Program of China (JCKY2019205A003).

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 14 April 2022 Accepted: 5 July 2022

Published online: 18 July 2022

References

1. S. Xu et al., A collaborative cloud-edge computing framework in distributed neural network. *EURASIP J. Wirel. Commun. Netw.* **2020**(1), 211 (2020)
2. H. Gao et al., TSMAE: a novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder. *IEEE Trans. Netw. Sci. Eng.* (2022). <https://doi.org/10.1109/TNSE.2022.3163144>
3. H. Gao et al., SDTIOA: modeling the timed privacy requirements of IoT service composition: a user interaction perspective for automatic transformation from BPML to timed automata. *Mob. Netw. Appl.* **26**(6), 2272–2297 (2021)
4. X. Ma et al., Real-time multiple-workflow scheduling in cloud environments. *IEEE eTrans. Netw. Serv. Manag.* **18**(4), 4002–4018 (2021)
5. Y. Yin et al., QoS prediction for service recommendation with features learning in mobile edge computing environment. *IEEE Trans. Cognit. Commun. Netw.* **6**(4), 1136–1145 (2020)
6. H. Gao et al., PPO2: location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems. *IEEE Trans. Intell. Trans. Syst.* (2022). <https://doi.org/10.1109/TITS.2022.3169421>
7. Y. Yin et al., Personalized APIs recommendation with cognitive knowledge mining for industrial systems. *IEEE Trans. Ind. Inf.* **17**(9), 6153–6161 (2021)
8. N.G. Paulter, Long-term repeatability of a TDR-based printed wiring board dielectric constant measurement system. *IEEE Trans. Instrum. Meas.* **47**(6), 1469–1473 (1998)
9. Q. Shi, O. Kanoun, Wire fault diagnosis in the frequency domain by impedance spectroscopy. *IEEE Trans. Instrum. Meas.* **64**(8), 2179–2187 (2015)
10. C.R. Sharma, C. Furse, R.R. Harrison, Low-power STDR CMOS sensor for locating faults in aging aircraft wiring. *IEEE Sens. J.* **7**(1), 43–50 (2007)
11. C. Lo, C. Furse, Noise-domain reflectometry for locating wiring faults. *IEEE Trans. Electromagn. Compat.* **47**(1), 97–104 (2005)
12. J. Tao et al., Design of aircraft cable fault diagnose and location system based on aircraft airworthiness requirement. *Procedia Eng.* **17**, 455–464 (2011)
13. O.D.J. Kuželka, Markov logic networks for knowledge base completion: a theoretical analysis under the MCAR assumption. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI, 2019*: p. 427.
14. Y. Zhang, W. Huang, Design of intelligent diagnosis system for teaching quality based on wireless sensor network and data mining. *EURASIP J. Wirel. Commun. Netw.* **2021**(1), 1–12 (2021)
15. S.C. Chu et al., Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification. *EURASIP J. Wirel. Commun. Netw.* **2020**(1), 1–5 (2020)
16. H. Gao, Z. Wang, S. Ji, *Large-Scale Learnable Graph Convolutional Networks* (ACM, New York, 2018)
17. M. Schlichtkrull et al., *Modeling Relational Data with Graph Convolutional Networks* (Springer, Cham, 2018)
18. D. Duvenaud et al., *Convolutional Networks on Graphs for Learning Molecular Fingerprints* (MIT Press, Cambridge, 2015)

19. N. Bandinelli, M. Bianchini and F. Scarselli. Learning long-term dependencies using layered graph neural networks. In The International Joint Conference on Neural Networks. 2010. IEEE
20. B. Hui, P. Zhu, Q. Hu, Collaborative graph convolutional networks: unsupervised learning meets semi-supervised learning. *Proc. AAAI Conf. Artif. Intell.* **34**(4), 4215–4222 (2020)
21. J. Liu, G.P. Ong, X.M. Chen, GraphSAGE-based traffic speed forecasting for segment network with sparse data. *IEEE Trans. Intell. Trans. Syst.* (2020). <https://doi.org/10.1109/TITS.2020.3026025>
22. Velickovi, P., et al., Graph attention networks. *arXiv preprint arXiv:1710.10903*. 2017.
23. H. Gao et al., A mutually supervised graph attention network for few-shot segmentation: the perspective of fully utilizing limited samples. *IEEE Trans. Neural Netw. Learn. Syst.* (2022). <https://doi.org/10.1109/TNNLS.2022.3155486>
24. R.A. Rossi, R. Zhou, N. Ahmed, Deep inductive graph representation learning. *IEEE Trans. Knowl. Data Eng.* (2018). <https://doi.org/10.1109/TKDE.2018.2878247>
25. B. Jamison, S. Olariu, On the homogeneous decomposition of graphs. *SIAM J. Discret. Math.* **8**(3), 448–463 (1992)
26. C. Shi et al., A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.* **29**(1), 17–37 (2016)
27. Y. Sun et al., PathSim: Meta path-based Top-K similarity search in heterogeneous information networks. *Proc. VLDB Endow.* **4**(11), 992–1003 (2011)
28. C. Shi et al., Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* (2017). <https://doi.org/10.1109/TKDE.2018.2833443>
29. Y. Long et al., Heterogeneous graph attention networks for drug virus association prediction. *Methods* (2021). <https://doi.org/10.1016/j.jymeth.2021.08.003>
30. Y. Dong, N.V. Chawla, A. Swami, *metapath2vec: Scalable Representation Learning for Heterogeneous Networks* (ACM, New York, 2017)
31. S. Yun et al., Graph transformer networks. *Adv. Neural Inf. Proc. Syst.* **32**, 11960–11970 (2019)
32. Y. Sun et al., PathSelClus: integrating Meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Trans. Knowl. Discov. from Data* **7**(3), 1 (2012)
33. Kipf, T.N. and M. Welling, Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*. 2016.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)