frontiers
in Energy Research

METHODS
published: 17 February 2021
doi: 10.3389/fenrg.2020.613331

Power System Network Topology Identification Based on Knowledge Graph and Graph Neural Network

Changgang Wang[1], Jun An[1,2]* and Gang Mu[2]

[1]Key Laboratory of Modern Power System Simulation and Control and Renewable Energy Technology, Ministry of Education (Northeast Electric Power University), Jilin, China, [2]School of Electrical Engineering, Northeast Electric Power University, Jilin, China

The automatic identification of the topology of power networks is important for the data-driven and situation-aware operation of power grids. Traditional methods of topology identification lack a data-tolerant mechanism, and the accuracy of their performance in terms of identification is thus affected by the quality of data. Topology identification is related to the link prediction problem. The graph neural network can be used to predict the state of unlabeled nodes (lines) through training on features of labeled nodes (lines) with fault tolerance. Inspired by the characteristics of the graph neural network, we applied it to topology identification in this study. We propose a method to identify the topology of a power network based on a knowledge graph and the graph neural network. Traditional knowledge graphs can quickly mine possible connections between entities and generate graph structure data, but in the case of errors or informational conflicts in the data, they cannot accurately determine whether the relationships between the entities really exist. The graph neural network can use data mining to determine whether a connection obtained between entities is based on their eigenvalues, and has a fault tolerance mechanism to adapt to errors and informational conflicts in the graph data, but needs the graph data as database. The combination of the knowledge graph and the graph neural network can compensate for the deficiency of the single knowledge graph method. We tested the proposed method by using the IEEE 118-bus system and a provincial network system. The results showed that our approach is feasible and highly fault tolerant. It can accurately identify network topology even in the presence of conflicting and missing measurement-related information.

Keywords: knowledge graph, power network, topology identification, graph neural network, knowledge inference

OPEN ACCESS

Edited by:
Chao Long,
Cranfield University, United Kingdom

Reviewed by:
Canbing Li,
Shanghai Jiao Tong University, China
Liang Chen,
Nanjing University of Information Science and Technology, China

****Correspondence:***
Jun An
anhuianjun@163.com

Specialty section:
This article was submitted to Smart Grids, a section of the journal Frontiers in Energy Research

Received: *02 October 2020*
Accepted: *21 December 2020*
Published: *17 February 2021*

Citation:
Wang C, An J and Mu G (2021) Power System Network Topology Identification Based on Knowledge Graph and Graph Neural Network. Front. Energy Res. 8:613331. doi: 10.3389/fenrg.2020.613331

INTRODUCTION

With access to new energy sources continuously increasing and the scale of power grids growing, the variability and complexity of the operation of power grids has increased drastically (Li et al., 2018). In recent years, several blackouts in areas across the world have caused substantial economic losses and led to adverse social impacts. The prevalent online security defense systems for power grids, which focus on modeling, simulation, and fault prediction, have been severely challenged. With the rapid development of power grid measurement systems and the increasing maturity of big data technology, the state cognition and control of grid operations based on operational information on power grids has emerged as the new model for their secure operation (Liu et al., 2018). In addition, the increasing uncertainties surrounding the use of renewable power has provided unprecedented challenges for ensuring the secure and economic operation of the current power systems (Li et al., 2019).

Understanding the topological structure of a power grid and laws of changes within it in a timely manner based on measurements is the basis for realizing the data-driven operation and control of power grids. Research on identifying the topology of a power network can be divided into several classes. Several studies have used the incidence matrix and the adjacency matrix constructed by using the switching state of the system to determine its connectivity and track topological changes (Zhu et al., 2011; Ma et al., 2014; Lourenco et al., 2015). Such methods are less tolerant of faults and conflicting telemetry data, and their effect depends entirely on the quality of remote signaling data. Other attempts have been made to identify the network topology based on graph theory and object-oriented technology (Song et al., 2005; Nian et al., 2008; Wang et al., 2009; Li et al., 2011; Yansheng et al., 2017). Such approaches involve using technology for the matrix-based identification of topology (Wang et al., 2019). They can compensate for the lower computational efficiency of the original process in terms of timeliness by optimizing the method, but are still reliant on remote signaling data for topology identification and have poor fault tolerance. Recently, the topology identification problem has been transformed into an optimal combination of measurement data problem. (Liu et al., 2019). It is a new attempt to identify network topology based entirely on data. The feasibility of this approach has been verified. Still, this method is not fault tolerant, and requires complete information concerning the active power. The determined optimal combination of data is directly affected when data are missing or when significant errors occur. Furthermore, this algorithm requires a large amount of redundant information as a supporting condition. As Power Management Units are not fully operational, owing to the massive investment required, the requisite supplementary data are difficult to obtain. In summary, current methods of topology identification depend on the quality of data. We thus need a fault-tolerant method of topology identification.

The use of knowledge graph technology has emerged in recent years in several fields, and has attracted widespread attention from both industry and academia (Liu and Wang, 2018). The knowledge graph is a mesh knowledge base of entities with attributes linked through relationships. Its value lies in organizing related information at minimal cost and generating useful knowledge. The topology of a power network is graph data reflecting node-to-node relationships. So the knowledge graph is an excellent vehicle for describing topological information (Park et al., 2019). However, information conflicts may occur in the knowledge graph due to errors or missing information. We thus need a data mining method to identify relationships that cannot be confirmed owing to the informational conflicts.

The graph neural network (GNN) is applicable mainly to non-Euclidean spatial data (Park et al., 2019). It has been applied to a variety of fields (Wu et al., 2018; Jing et al., 2020; Li et al., 2020; Xie et al., 2020; Yu et al., 2020). Multiple types of graph neural networks have been derived for different problems (Zhang et al., 2019; Nikolentzos et al., 2020; Su et al., 2020; Wu et al., 2020). Because the knowledge graph is a kind of graph data, a combination of the GNN and the knowledge graph can be used to solve all types of knowledge graph-related problems. For example, some researchers have modeled the knowledge graph with the GNN (Nathani et al., 2019). The GNN-based approach can adequately capture information on complex and hidden patterns in ternary neighborhoods. It can also achieve the relational complementation of the knowledge graph, in contrast with knowledge-based reasoning that uses individual ternary relationships. A GNN-based entity alignment scheme has been proposed (Cao et al., 2019), and it has been experimentally shown to yield highly consistent data that improve the quality of the knowledge graph on multiple datasets.

In the context of power network topology, a knowledge graph can indicate relationships between entities in the network. The graph neural network can serve as a technical tool for analyzing graph data, mining, and reasoning for relationships among entities in the knowledge graph. It also allows us to consider missing information and determine the correctness of the data, significantly improving the quality of the knowledge graph. Thus, the characteristics of the knowledge graph and the graph neural network are fully compatible with the requirements of topological recognition.

This paper proposes a method of topological identification based on the knowledge graph and the graph neural network. First, we set-up an entity–relationship–entity and entity–attribute/attribute value information triad based on remote signaling data, telemetry data, and a database of component information to construct an informational knowledge base. Second, based on the graph neural network, we determine the relationships that cannot be established owing to conflicting information. Finally, based on knowledge inference, we generate the topology of the power network and track changes in it. The results of tests show that the proposed method is fault tolerant and can accurately determine the network topology of a simulated network as well as a real power grid.

DEFINITION AND STRUCTURE OF KNOWLEDGE GRAPH

Definition of Knowledge Graph

A knowledge graph is a structured semantic knowledge base used to describe concepts and their interrelationships in the physical world in symbolic form. The basic unit is the "entity–relation–entity" triad, and the entity and its associated attribute–value pairs are interconnected through relationships to form a networked knowledge structure.

Based on the above definitions, we can draw the following conclusion:

(1) Essentially, the knowledge graph is a semantic network that reveals the relationships between entities.
(2) The research value of the knowledge graph lies in the fact that it is a reticular knowledge base built based on data and

information. It organizes the information into usable knowledge at minimum cost.

(3) The applicative value of the knowledge graph is that it can change the existing method of information retrieval. On the one hand, retrieval is achieved by reasoning. On the other hand, it can graphically display the structured knowledge after classification.

Architecture of the Knowledge Graph

The architecture of the knowledge graph consists of a data layer and a schema layer, as shown in **Figure 1**. The data layer is toward the logical architecture of the knowledge graph and the schema layer toward its technical architecture. The primary function of the data layer is the extraction of data and information to generate relationships inherent within the latter. The schema layer builds on top of the data layer and is an essential part of processing and analyzing data to generate knowledge, including such means as information fusion and processing, and creating knowledge based on inference.

METHODS OF CONSTRUCTING POWER GRID TOPOLOGY-ORIENTED KNOWLEDGE GRAPHS

The power information system records a large number of structured or semi-structured data. These data are descriptions of the measured attributes and results, and the determined entity attributes. All these data provide the necessary information units to construct knowledge graphs of the topology of the power network.

Methods of Entity–Property Construction

A large number of fields in databases recorded by the power management information system are deconstructed. The measured set of items of entity information O is thus obtained. It contains all information on nodal entities and line entities, given as:

$$O = \left\{N_i, N_j, L_{ij}\right\} \tag{1}$$

where N_i and N_j are node information units and L_{ij} is a line information unit.

The information provided by each entity corresponds to its entity attribute and attribute value. For example, an entity node (N_i) section corresponds to the description node inject telemetry (K_{ni}) and node inject telemetry (P_{ni}). Remember that an entity node (N_i) has the following attributes:

$$v_i = (K_{ni}, P_{ni}) \tag{2}$$

The resulting set of properties corresponding to the entity is as follows:

$$V = \{v_i\} \tag{3}$$

Entity O and attribute V correspond in a one-to-one manner. The expression for entity–attribute Q is as follows:

$$Q = (O, V) \tag{4}$$

Methods of Constructing Entity Relationships

Relationships are key to the knowledge graph because they require its use to solve a number of problems. In general, they make it possible to further parse the entity correspondences in the entity information set O, and to organize the entity–relationship information. The expression for the relationship is given as:

$$R_{ni,lij} = \left(N_i, L_{ij}\right) \tag{5}$$

where N_i represents node i, L_{ij} represents the line ij directly connected to node i, and $R_{ni,lij}$ is the value of the relationship between them. 1 indicates that they are directly connected and 0 indicates that they are not.

Methods for Constructing Knowledge Graph Triangles

In this paper, entity–relation–entity and entity–property–attribute (value) are transformed into an integrated entity (attribute–attribute

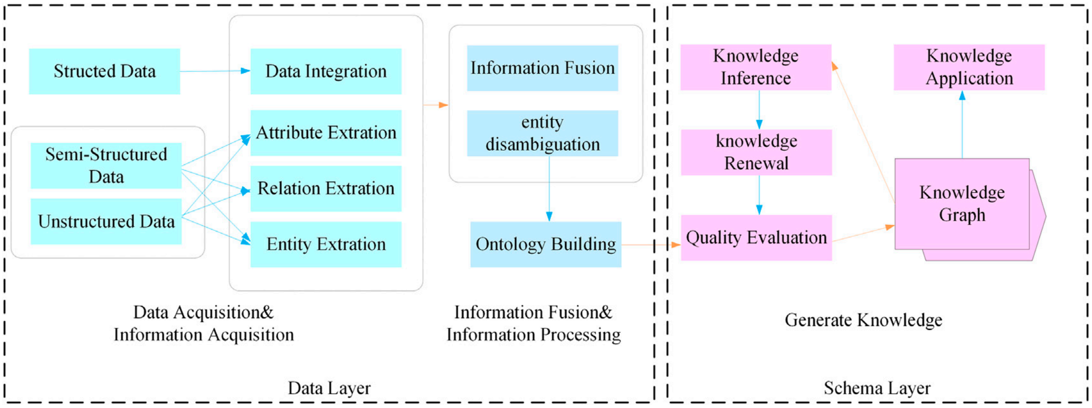

FIGURE 1 | The architecture of the knowledge graph.

value)–relation–entity (attribute–attribute value) triad to form a knowledge base for generating topologies. The expression is given as:

$$G = (O, V, R) \tag{6}$$

where G denotes the set of ternary groups, O the set of entities, and each object is unique. V denotes the set of attributes, including the number of nodes and states of line switches, the magnitude of node-injected power, the first-end power of the line, and the line first-end power difference. R denotes the set of relationships of direct correlations between entities.

GRAPHICAL NEURAL NETWORK

Traditional neural networks have had great success in extracting features from spatial Euclidean data, but are deficient in processing non-Euclidean spatial data. In recent years, researchers have learned from the ideas of convolutional networks, cyclic networks, and deep autoencoders. They have used this knowledge to design a neural network structure, called the graph neural network, for processing graph data.

A graph is composed of many vertices and edges, usually represented and stored using an adjacency matrix. The spatial features in graph data have the following characteristics:

(1) Node features: Each node has certain features.
(2) Structural features: Each node in the graph data has certain structural features.

Thus, there is a connection between pairs of nodes. In general, graph data should consider both node and structural information, and a graph neural network should be able to automatically learn both the characteristics of a node and node-to-node correlation-related information.

GCNs are the first graph-based networks to apply the simple convolution operation to image processing for graph data processing. This paper explains the principles of graph convolutional neural networks mainly from the perspective of spatial construction.

A grid can be substituted with a graph $g = (\Omega, A)$, where Ω is defined as a discrete set of nodes $\Omega \in \mathbb{R}^{N\times 1}$, and W denotes the set of edge weights $W \in \mathbb{R}^{N\times N}$. A straightforward way to represent edge weights on W is to set a threshold $\delta > 0$. Nodes adjacent to Node j can be represented as:

$$N_i = \left\{ i \in \Omega : W_{ij} = \delta \right\} \tag{7}$$

As shown in **Figure 2**, we want to convolve node 6, and adjacent to it are nodes 1 and 5 (including node 6 itself); that is $N[6]$. Therefore, a convolution of node 6 can be expressed as $W_{1,6}x_1 + W_{5,6}x_5 + W_{6,6}x_6$, and x denotes the features of every node, and w denotes the weight of the convolution.

o_v, as the convolution of the node v, can be expressed as:

$$o_v = \sum_{u \in N[v]} W_{u,v} x_u \tag{8}$$

The input node feature may be a vector with dimensions D. A single convolution operation may contain multiple convolution kernels with dimensions d. We convolve every dimension of the input feature and sum it to obtain the convolution of node v. The formula is as follows:

$$o_{v,j} = \sum_{i=1}^{D} \sum_{u \in N[v]} w_{i,j,u,v} x_{u,i}, j = 1 \cdots d \tag{9}$$

The basic idea of the GCN is to reduce the dimensionality of a node's high-latitude adjacency information in the graph to a low-dimensional vector representation by aggregating feature-related information from adjacent nodes. It can aggregate the global information of the graph to represent features of the nodes. The process of updating the feature-related information of the node is shown in **Figure 3**.

The convolution operator can be defined as (Kipf and Welling, 2017):

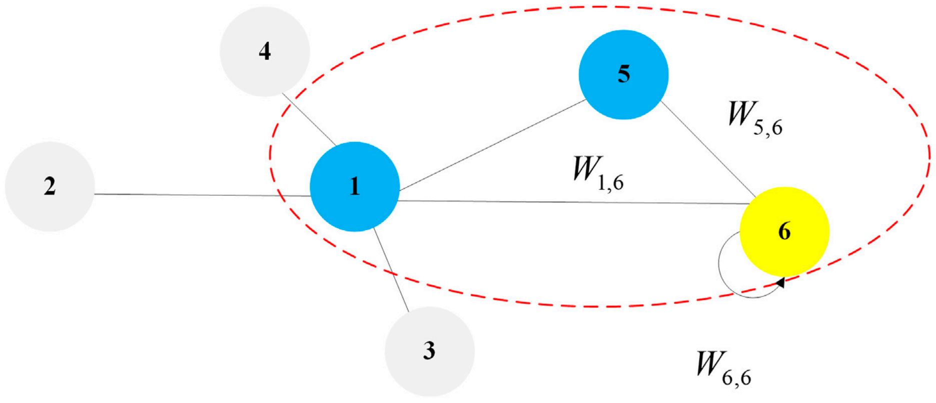

FIGURE 2 | Schematic diagram of spatial convolution for node 6.

$$h_v = f\left(\frac{1}{|n_{(v)}|}\sum_{u \in N[v]} WX_u + b\right), \forall v \in \Omega \tag{10}$$

where h_v is the linearly transformed feature value of node v after aggregating features of neighboring nodes, u is the set of nodes adjacent to v, $n_{(v)}$ is the number of nodes neighboring v, W is the weight, and b is the bias. The formula for passing features through multiple hidden layers is given as:

$$h_v^{k+1} = f\left(\frac{1}{|n_{(v)}|}\sum_{u \in N[v]} W^k h_u^k + b^k\right), \forall v \in \Omega \tag{11}$$

The matrix form of the above equation is as follows:

$$H^{(k+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(k)}W^{(k)}\right) \tag{12}$$

The adjacency matrix A does not contain information on the nodes themselves. To solve this problem, given $\tilde{A} = A + I$, I is a unit array. Define the degree matrix D, $D \in \mathbb{R}^{N\times N}$, that is a diagonal matrix characterizing the degree of connectivity of a node to other nodes $\overline{D_{ii}} = \sum \overline{A_{ij}}$. Thus, the normalized $\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is used as a medium to pass information in the topological space to solve the diagonal problem so that the training covers information on the nodes. $H^{(k)} \in \mathbb{R}^{N\times F}$ is the feature matrix after k-layer activation, $H^{(0)} = X$, and $W^{(k)}$ is the learnable parameter matrix of the kth layer.

In short, compared with the basic structure of a neural network, the MLP, the feature matrix multiplied by the weight matrix $H = \sigma(XW)$, adds an adjacency matrix to the graph neural network $H = \sigma(AXW)$.

The core framework of a graph neural network consists of two matrices: 1) the adjacency matrix A, $A \in \mathbb{R}^{N\times N}$, where N is the number of nodes, and 2) the feature matrix X, $X \in \mathbb{R}^{N\times F}$, where N represents the number of nodes in the graph. F is the number of dimensions of the feature of each node. Moreover, the output matrix is Z, $Z \in \mathbb{R}^{N\times C}$, and C represents the number of convolutional kernels used to implement the classification. The graph neural network is designed to maximally match the input and the output, and continually train itself to classify the input data. **Figure 4** shows the flow of the graph neural network.

PROPOSED METHOD FOR IDENTIFYING POWER NETWORK TOPOLOGY

Constructing Overall Topology of Power Grid

Through information extraction, elements of knowledge such as entities, relations, and properties can be extracted from the original corpus to obtain a series of basic factual expressions.

The relationship between entities can be extended according to knowledge inference. For example, the node Na in the entity information set O directly connects to line L_{ab}, and node N_b in the set is also directly connected to L_{ab}. The relationship can be extended to N_a-L_{ab}-N_b. Multiple entity–relation unit elements can form a topology containing all entity–relation information.

Identifying Power Network Topology

The above is not identical to knowledge. To obtain a structured and networked knowledge system, it is necessary to carry out knowledge processing and reasoning. Knowledge processing and knowledge reasoning are the essential means and critical parts of the construction of a knowledge graph. In practice, remote signaling data may be mixed with incorrect information owing to the possible existence of the blade gate relation. Telemetry data may also be combined with errors, such that the resulting data do not comply with the operating rules of the power system. Ambiguity may thus persist within the entity–relation–attribute triad, which is required for the most critical step in topological identification based on the knowledge graph. The authenticity of the entity–relation units is thus verified through the data mining method of graph neural networks.

The main problem in determining power network topology is to identify the connectivity of the lines based on their properties (values). In applying the graph neural network, we use the relationship between entities in the knowledge graph (lines) as "vertices" and the entities (nodes) as "edges." In this way, the connectivity of lines is directly generated by line training, which helps to avoid the conventional approach of first determining the nodes and then identifying connectivity. The follow chart of the proposed method is shown in **Figure 5**.

The specific steps are as follows:

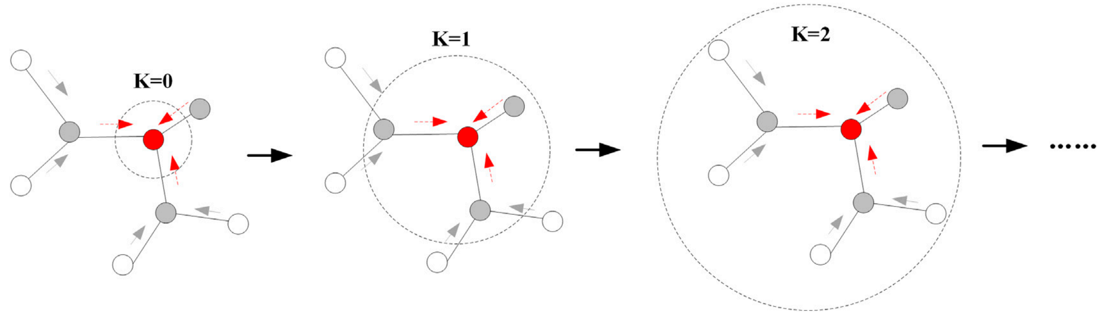

FIGURE 3 | Schematic diagram of node feature update based on graph convolutional neural network.

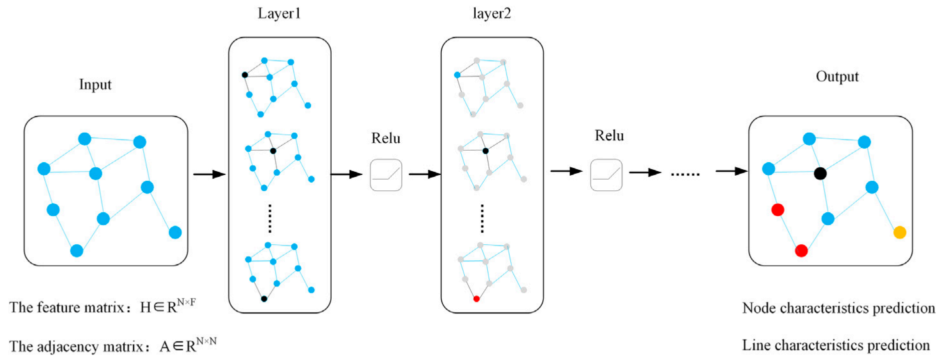
Layer1
layer2
Input
Relu
Relu
......
Output
The feature matrix：$H\in R^{N\times F}$
The adjacency matrix：$A\in R^{N\times N}$
Node characteristics prediction
Line characteristics prediction

FIGURE 4 | Flowchart of graph neural network.

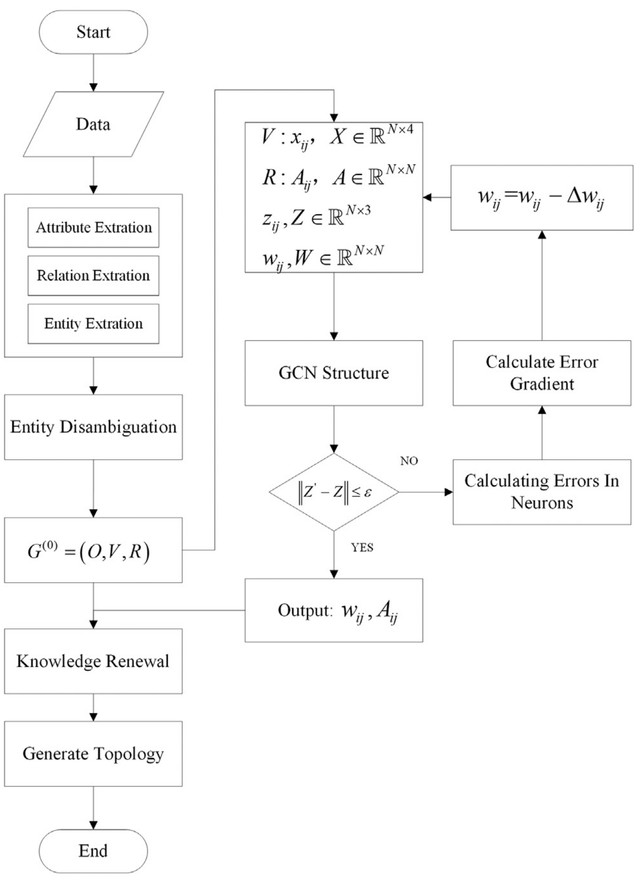
Start
Data
Attribute Extration
Relation Extration
Entity Extration
Entity Disambiguation
$G^{(0)}=(O,V,R)$
Knowledge Renewal
Generate Topology
End
$V: x_{ij},\ X\in\mathbb{R}^{N\times4}$
$R: A_{ij},\ A\in\mathbb{R}^{N\times N}$
$z_{ij}, Z\in\mathbb{R}^{N\times3}$
$w_{ij}, W\in\mathbb{R}^{N\times N}$
$w_{ij}=w_{ij}-\Delta w_{ij}$
GCN Structure
Calculate Error Gradient
$\|Z^{'}-Z\|\le\varepsilon$
NO
Calculating Errors In Neurons
YES
Output: w_{ij}, A_{ij}

FIGURE 5 | The flowchart of the proposed method.

Step 1. Extract the attributes and attribute values of edges in the knowledge graph that contain the line input power P_{ij} and the line output power P_{ji}, and the head switch statuses of the line S_{ij} and S_{ji} to form a description of the characteristic matrix of the line:. P_{ij} or P_{ji} is the measured power, and S_{ij} and S_{ji} are the descriptions of the on and off states of the line: on is 1 and off is 0.

TABLE 1 | comparison of variables and parameters.

Node i	Node j	Line input power/ MW	Line output power/ MW	Line head switching status	Line end switching status
8	30	0	0	0	0
48	49	0	0	0	0
17	18	0	0	1	1
21	22	−42.837	43.256	0	0
34	36	0	−30.162	1	1
76	77	−61.150	0	1	1
84	85	0	36.791	1	1
92	100	31.479	0	1	1
105	107	26.755	−26.347	0	1
114	115	1.358	−1.358	1	0

Step 2. P_{ji} and P_{ij} in x_{ij} are normalized to form $\widehat{x}_{ij}$ to ensure that the eigenvalues $\widehat{x}_{ij}$ are in the interval [-1,1], such that all $\widehat{x}_{ij}$ are aggregated into X, $X\in\mathbb{R}^{N\times4}$, and N is the total number of lines.

Step 3. Define z_{ij} as the line category identifier, and make all aggregates into Z, which contains all line categories for the graph neural network training comparison.

Step 4. Extract all entity relations of the knowledge graph $G=(O,V,R)$, that is, the node–line relationship, to form an adjacency matrix A, with a line as the vertex and a node as the edge, to describe the direct connection between lines. A_{ij} represents whether there is a common node between lines i and j.

Step 5. Set $IT_{\max}$ to the maximum number of iterations. Substitute the adjacency matrix A and the feature matrix X into the feature transfer formula, calculate $Z^{'}$. $\|Z^{'}-Z\|$ as a loss function, and modify the weight matrix continuously. After iterative computation, use the weight matrix W of each hidden layer, on behalf of the graph neural network after self-training rules of decision, directly to determine line connectivity.

Step 6. After connectivity determination, update the adjacency matrix $A^{'}$. Finally, generate the topology by deducing the correlation between the internal line and the line according to depth search to analyze the generated topology and determine whether isolated nodes and islands are in operation.

Algortihm 1 | Overall training and testing process

Input:

Graph $G(V, \varepsilon)$; Input features $\{X_v, \forall v \in V\}$; randomly initialized network weight W; non-linearity σ; the maximum number of iterations $IT_{\max}$; GCN structure;

Output:

Network model , the weight value W, Generate topology

```
    Function:
    Calculating the graph structure $G$;
    Normalize input features;
    Integrate features to get GCN input through $G$;
    Integrate features to get corresponding labels.
    for range of $IT_{\max}$ do
    Calculate network through input;
    Calculate loss through network output and labels;
    Do backpropagation algorithm through optimizer, update network weight $W$
    end
Calculate network output;
Do anti-normalization for the network output
generate topology;
Save network model;
```

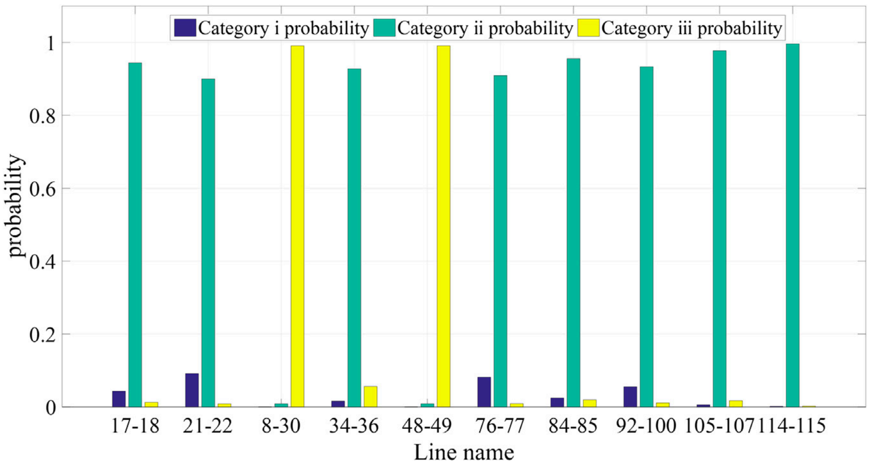

FIGURE 6 | Histogram of probability of line determination for a disconnected line or missing information.

EXPERIMENT AND RESULTS

Experimental Results on a Test Sample

The IEEE 118-node system containing 186 lines was used for our experiment. The initial active power of the nodes was obtained by the MATPOWER software simulation. We set-up two disconnected lines: 8–30 and 48–49. The lines 17–18, 21–22, 34–36, 76–77, 84–85, 92–100, 105–107, and 114–115 contained conflicting information. The remaining lines were normal. Details of the information in the lines are shown in **Table 1**. Algorithm 1 shows the overall training and testing processes.

The above sample was selected to demonstrate the effectiveness of the proposed method. We followed the flow of Algorithm 1 in calculating the sample data, and then used the result of the judgment of this sample with the original features of the line. The probability of categorization and the number of line characteristics that determine the presence of conflicting information and "off" lines are shown in **Figure 6** and **Table 2**.

It is clear that only lines 8–30 and 48–49 had a high probability of falling into category iii in the above diagram. According to the results of training of the graph neural network, these two lines were judged to be disconnected, consistently with the original perturbation.

According to the probability distribution, lines 27–115, 114–115, 68–116, 12, 117, 54–55, 80–81, 75–118, and 76–118 were all judged to be connected but had conflicting information. The results were consistent with the original settings. We updated the knowledge graph based on the above results of identification. A comparison of the results of topological identification is shown in **Figure 7**.

Experimental Results on the IEEE 118-Node System

We used an IEEE 118-node system containing 186 lines. The initial active power of the nodes was obtained by MATPOWER software simulation. A 1% random error obeying a Gaussian

TABLE 2 | Probability distribution of the presence of conflicting information and disconnected lines in various classifications.

Node *i*	Node *j*	Category i probability	Category ii probability	Category iii probability
8	30	0.0051	0.0163	**0.9786**
48	49	0.0051	0.0163	**0.9786**
17	18	0.0012	**0.8854**	0.1134
21	22	0.0009	**0.9987**	0.0004
34	36	0.0001	**0.9996**	0.0004
76	77	0.0003	**0.9969**	0.0028
84	85	0.0055	**0.9772**	0.0172
92	100	0.0016	**0.9963**	0.0021
105	107	0.0019	**0.9979**	0.0002
114	115	0.0014	**0.9984**	0.0002

The number in bold is the value with the largest probability distribution for each line. It indicates that the lines are grouped into the corresponding categories.

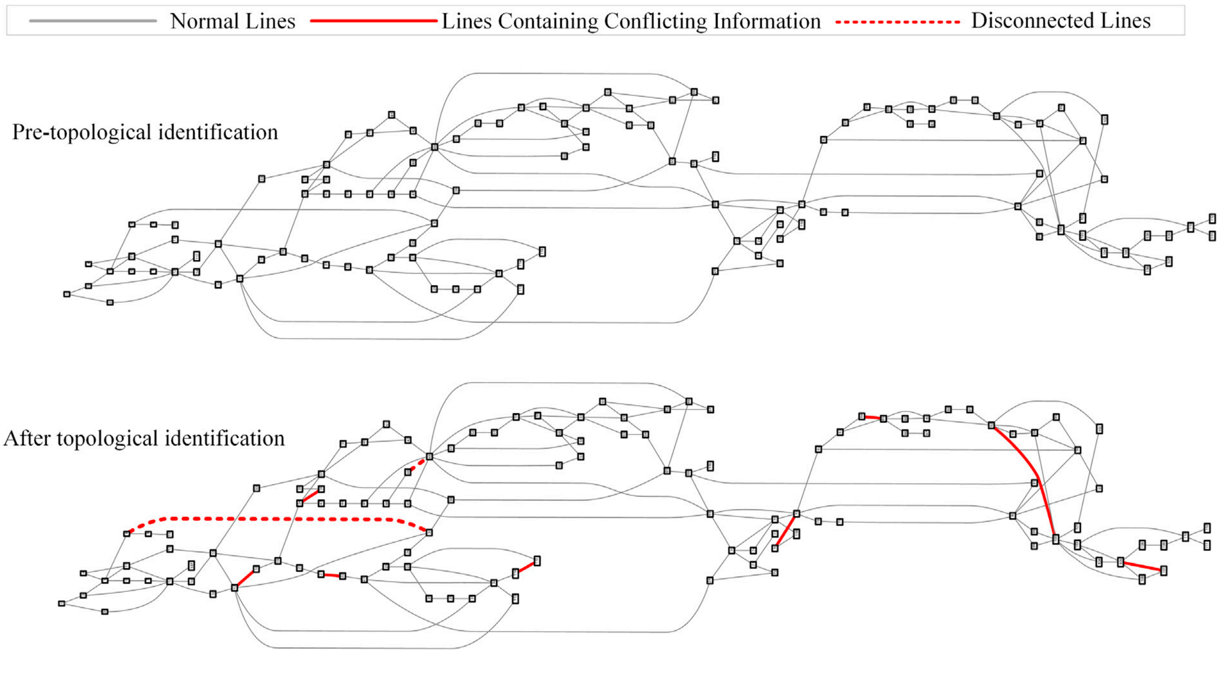
Normal Lines
Lines Containing Conflicting Information
Disconnected Lines
Pre-topological identification
After topological identification

FIGURE 7 | Comparison of results of topological identification.

distribution for active node power was mixed in to generate 1,000 training samples. For each sample, two lines were randomly disconnected, and eight lines were made to randomly miss some attribute values. We formed an information base for the topology of the power grid according to the construction of the knowledge graph, and organized the data and relationships in the node, line feature matrix, and adjacency matrix. The feature matrices of 900 training samples were labeled as follows: lines in label category i represented connected lines, those in label category ii represented lines with conflicting information in the feature matrix despite being related lines, and lines in label category iii represented disconnected lines. One hundred samples were retained as unlabeled samples.

Seven hundred labeled samples were used as input to the graph neural network for training and another 200 were used to compare the ingested topology with the generated topology through the rules learned from the graph neural network.

For each sample, the results of identification of line connectivity could be organized into a 186 × 1 column vector, and the number of dimensions of the matrix of results for 200 samples were 186 × 200. By marking different categories with different colors, a map of the results of connectivity was obtained for all samples. We statistically determined the determinations of the remaining 199 samples. The results are shown in **Figure 8**.

In the figure above, the abscissa represents the number of test samples and the ordinate the line number. The blue blocks represent lines judged to belong to category ii and the red blocks represent those judged to be in category iii. Each sample contained two category iii lines and eight category ii lines. Compared with the initial topology, the graph neural network method accurately identified the disconnected lines.

By using the rules learned from the graph neural networks, we identified the topologies of the remaining 100 unlabeled samples. The results are shown in **Figure 9**.

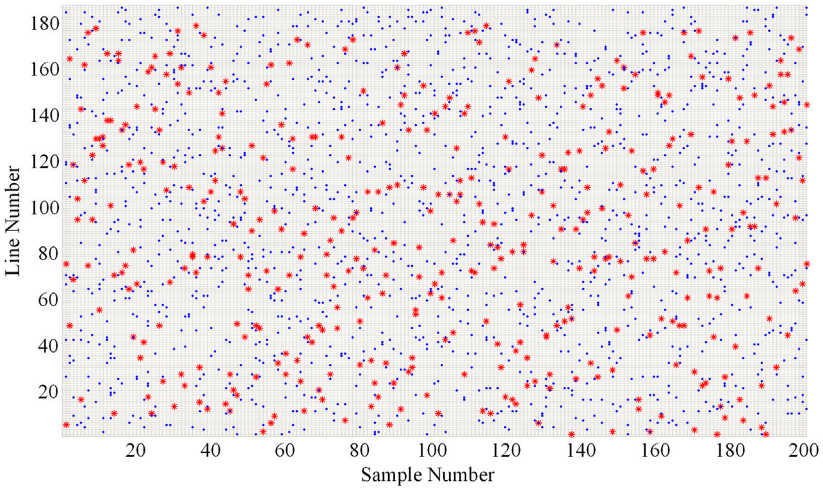

FIGURE 8 | Scatter plot of results of identification of test samples.

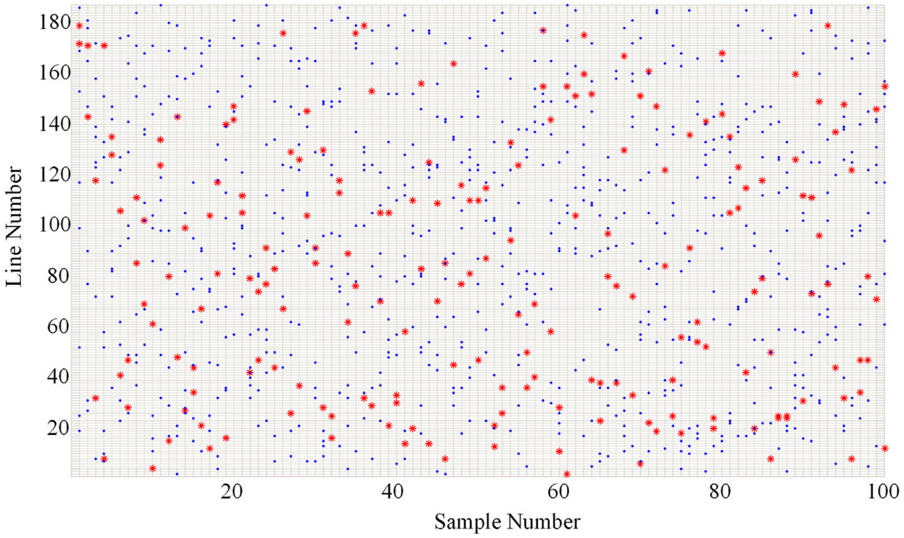

FIGURE 9 | Scatter plot of results of identification of unlabeled samples.

As shown in the figure above, we were able to identify the operating status of each line. Each unlabeled sample contained two Category iii lines and eight Category ii lines, which is consistent with the original assumption.

For the samples remaining as labels, the right decision is made to compare the decision results with their original feature quantities, and the results are consistent.

To further verify the proposed method, we selected the above 100 unlabeled samples as test samples. The method used for comparison with the proposed one was a topology identification method based only on knowledge graphs. Both methods are applied to the same data sample.

Each sample contained 184 connected lines and two disconnected lines. Eight of the connected lines were partially missing information. As is shown in **Figure 8**, the proposed method accurately identified the line connections and determined the classification of each line. We thus determined that there were 18,400 connected lines and 200 disconnected lines in the 100 test samples; 800 of the connected lines had missing partial information, which was consistent with the original settings. The comparison of the results with those of the method for topology identification based only on knowledge mapping is shown in **Figure 10**.

The method for topology identification based only on knowledge mapping accurately identified disconnected lines, but its accuracy of identifying connected lines was 95.65%. This is mainly because knowledge-based topology identification was less fault tolerant than the proposed method and struggled to deal with missing information.

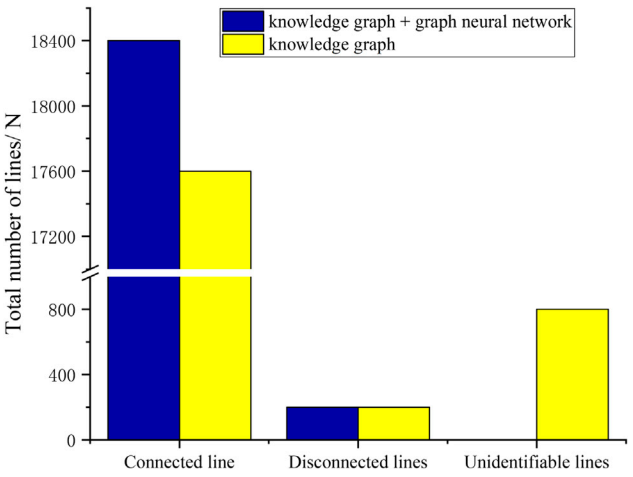

FIGURE 10 | Results of topological identification of the two methods.

In summary, the proposed method for topology identification based on a combination of the knowledge graph and the graph neural network can determine the state of line connectivity, identify topological changes, and has strong fault tolerance. Even in the case of conflicting information, it can identify the topology of the power network, and is thus superior to the traditional method of recognition based only on the knowledge graph.

Experimental Results on a Provincial Network

The proposed method was applied to a provincial network consisting of 132 nodes and 181 lines. The data were recorded at intervals of 1 min. We combined the knowledge graphs and the graph neural networks to identify the topology of the power network.

Historical operational data provided a sufficient number of samples to train the graph neural network, thus enabling it to make more accurate decisions. Nevertheless, with increasing numbers of training samples, the number of dimensions of the

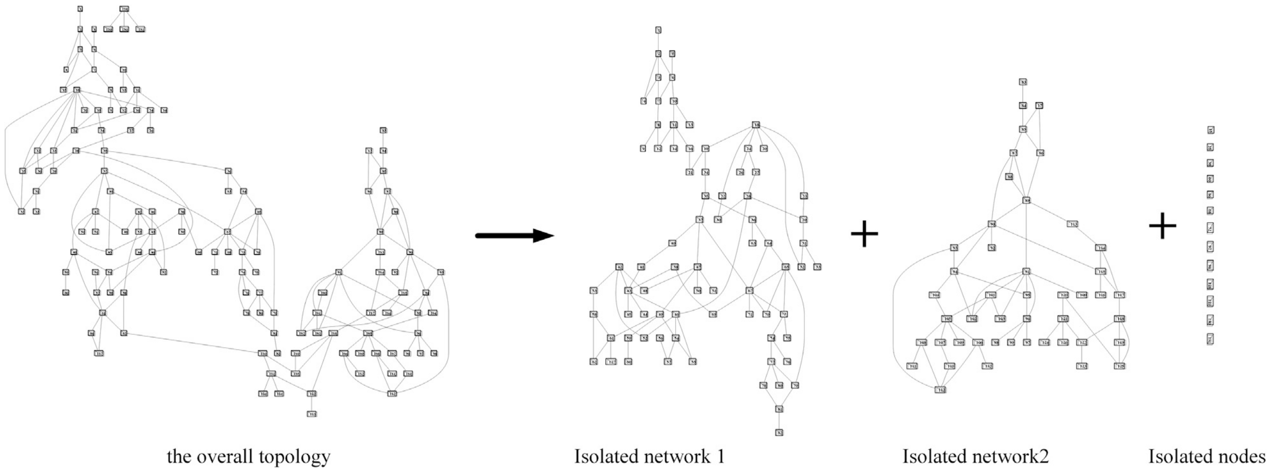

FIGURE 11 | Network structure after topology identification.

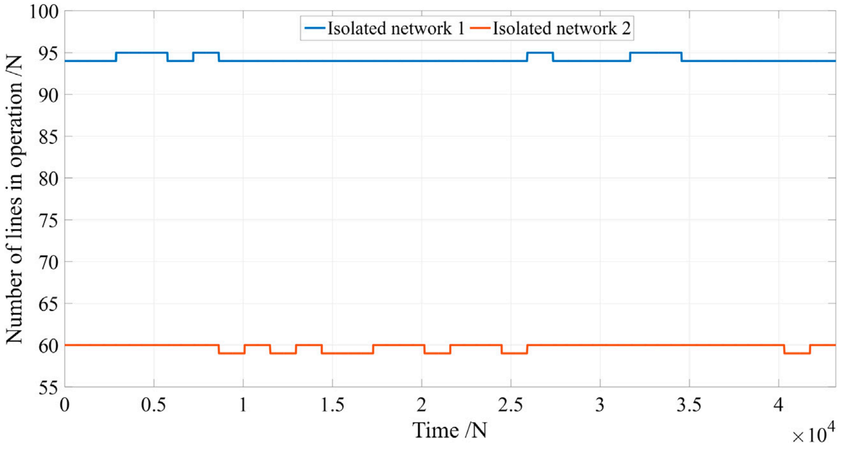

FIGURE 12 | Changes in the number of lines in operation on the isolated networks in the time series.

adjacency matrix A increased from the original $N \times N$ to the original $\tilde{N} \times \tilde{N}$, $\tilde{N} = N*(t+1)$. Considering that the training time increased with the amount of input data, we trained the samples with a window length of 10 min and a step length of 10 min, used the number of line features in period t as input, and determined the network topology in period *t*+1 based on the results of training.

Based on one-month data from the power grid, we found that the provincial network comprised two orphaned network topologies and several isolated nodes. The details are shown in **Figure 11**.

As shown in **Figure 6**, the full-time network consisted of 13 remote nodes and two isolated networks, with a total of 119 nodes in continuous operation and 153–155 lines involved. The structural information of the network is shown in **Table 3**.

We verified that 13 of the isolated nodes were planned nodes that had not yet been put into use in the given month, which is consistent with the results of topology identification. Isolated network 1 contained 75 nodes with 94–95 lines in operation, of which lines 28–46, 42–52, and 65–67 had pitching and cutting variations. The remaining lines in isolated network 1 were in stable operation. Isolated network 2 contained 44 nodes and 59–60 lines in operation, of which lines 91–96 and 118–125 had pitching and cutting variations. The number of lines in operation in chronological order in two isolated networks are shown in **Figure 12**.

The results show that the proposed method is feasible. It can accurately identify the state of line switching, generate network topology, update the knowledge graph, and improve its quality. It also provides a good basis for analyzing the islanding operation, demodulation control, and the operating state and control of power grids.

TABLE 3 | Information on the composition of the network structure.

Network structure	Node name	Line name
Isolated nodes	15','25','39','60','66','68','72','97','98','128','129','130','131'	
Isolated network 1	1','2','3','4','5','6','7','8','9','10','11','12','13','14','16','18','19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','36','37','38','40','41','42','43','44','45','46','47','48','49','50','51','52','53','54','55','56','57','58','59','61','62','63','64','65','67','69','70','71','73','74','75','76','77','78','79','80','81','82','127'	1-2','2-3','2-4','2-6','3-4','3-7','5-6','6-7','6-10','7-8','8-9','8-12','10-11','10-13','11-12','11-14','11-16','16-21','18-19','18-20','18-22','18-23','18-24','18-32','19-21','19-34','22-30','23-29','24-26','24-27','27-28','28-29','28-30','28-46','29-31','31-32','31-33','34-35','35-36','35-37','36-63','36-64','37-40','37-47','37-67','38-43','38-69','40-41','41-43','42-43','42-48','42-52','42-53','43-44','43-45','43-46','46-52','46-54','46-55','47-48','47-49','47-50','47-51','49-54','49-59','49-61','53-58','54-55','54-57','56-57','58-61','58-62','61-62','61-127','64-65','64-67','65-67','65-70','65-73','65-79','67-69','67-70','67-71','67-73','73-74','73-75','74-76','74-77','77-78','77-79','77-80','78-81','79-81','81-82'
Isolated network 2	17','83','84','85','86','87','88','89','90','91','92','93','94','95','96','99','100','101','102','103','104','105','106','107','108','109','110','111','112','113','114','115','116','117','118','119','120','121','122','123','124','125','126','132'	17-85','17-86','83-84','84-85','85-86','85-87','86-89','87-88','87-89','87-90','88-89','89-90','89-91','89-113','90-92','90-93','90-94','90-115','91-95','91-96','91-100','91-101','91-102','91-103','93-94','93-112','94-95','94-102','94-104','95-96','96-99','101-102','101-105','104-105','105-106','105-107','105-108','105-109','106-112','106-132','107-110','108-111','108-112','110-112','113-114','114-115','114-117','115-116','115-117','117-118','118-119','118-122','118-125','119-125','120-121','120-125','121-122','121-124','121-126','122-123'

CONCLUSION

The main contribution of this paper is the proposal of a method based on a knowledge graph and a graph neural network. This method transforms the traditional problem of topology identification into one of inferring the state of line connections based on graphical data for topology identification.

The proposed method is distinct from methods that used knowledge graphs only for topology identification because it contains an additional process for inferring conflicting information. After obtaining the overall topology covering all entities in the network, we performed knowledge inference on contradictory information based on graph neural networks. We then determined the line connectivity and updated the entity–relation information in the graph. In this way, we compensated for the deficiency of topological identification based only on the knowledge graph while making substantial gains in identifying topology and tracking changes in it.

We experimentally demonstrated that our proposed method is fault tolerant, unlike traditional methods, and correctly identifies line connectivity even in the case of informational conflicts.

We tested the proposed approach on an operational power grid to show that it can identify network topology, isolated island operations in the network, and can track changes in topology. This can yield valuable information for analyzing power system operations.

The graph neural network expands the training samples using historical data. The dimensionality of the adjacency matrix increases exponentially, which inevitably reduces training efficiency. Optimizing the graph neural network to improve efficiency should be investigated in future work.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

CW: Data curation, Writing–Original draft preparation and Writing–Reviewing JA: Conceptualization and Methodology GM: Supervision and Editing.

FUNDING

This work was supported by the National Natural Science Foundation of China (No. 51877034).

REFERENCES

Cao, Y. X., Liu, Z. Y., Li, C. J., Liu, Z. Y., Li, J. Z., and Chua, T. S. (2019). "Multi-channel graph neural network for entity alignment," in 57th annual meeting of the association for computational linguistics, Florence, Italy, August 26, 2019 (Stroudsburg, PA: Acl 2019), 1452–1461.

Jing, Y., Wang, J., Wang, W., Wang, L., and Tan, T. (2020). Relational graph neural network for situation recognition. *Pattern Recogn.* 108, 107544. doi:10.1016/j.patcog.2020.107544

Kipf, T. N., and Welling, M. (2017). "Semi-supervised classification with graph convolutional networks," in 5th International Conference on Learning Representations, Toulon, France, April 24, 2017–April 26, 2017.

Li, F., Liu, T., and Jiang, D. (2011). Distribution network reconfiguration with multi-objective based on improved immune algorithm. *Dianwang Jishu/Power Syst. Technol.* 35 (7), 134–138. doi:10.13335/j.1000-3673.pst.2011.07.028

Li, M., Ma, Z., Wang, Y. G., and Zhuang, X. (2020). Fast Haar transforms for graph neural networks. *Neural Network.* 128, 188–198. doi:10.1016/j.neunet.2020.04.028

Li, Y., Wang, J., Zhao, D., Li, G., and Chen, C. (2018). A two-stage approach for combined heat and power economic emission dispatch: combining multi-objective optimization with integrated decision making. *Energy* 162, 237–254. doi:10.1016/j.energy.2018.07.200

Li, Y., Yang, Z., Li, G., Zhao, D., and Tian, W. (2019). Optimal scheduling of an isolated microgrid with battery storage considering load and renewable generation uncertainties. *IEEE Trans. Ind. Electron.* 66 (2), 1565–1575. doi:10.1109/tie.2018.2840498

Liu, D., Zhang, R., Li, B., Yun, Z., Yang, H., and Li, Z. (2019). Research on reverse recognition method of power system topological structure based on state measurement. *Zhongguo Dianji Gongcheng Xuebao/Proc. Chin. Soc. Electr. Eng.* 39 (24), 7107–7117. doi:10.13334/j.0258-8013.pcsee.182454

Liu, Y., Zhang, N., and Kang, C. (2018). A review on data-driven analysis and optimization of power grid. *Dianli Xitong Zidonghua/Autom. Electr. Power Syst.* 42 (6), 157–167. doi:10.7500/AEPS20170922003

Liu, Z., and Wang, H. (2018). Retrieval method for defect records of power equipment based on knowledge graph technology. *Dianli Xitong Zidonghua/Autom. Electr. Power Syst.* 42 (14), 158–164. doi:10.7500/AEPS20180103007

Lourenco, E. M., Coelho, E. P. R., and Pal, B. C. (2015). Topology error and bad data processing in generalized state estimation. *IEEE Trans. Power Syst.* 30 (6), 3190–3200. doi:10.1109/Tpwrs.2014.2379512

Ma, J., Zhang, Y., Ma, W., and Wang, Z. (2014). Power network topological analysis based on incidence matrix notation method and loop matrix. *Dianli Xitong Zidonghua/Autom. Electr. Power Syst.* (12), 74–80. doi:10.7500/AEPS20131212003

Nathani, D., Chauhan, J., Sharma, C., and Kaul, M. (2019). "Learning attention-based embeddings for relation prediction in knowledge graphs," in 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, ACL, July 28, 2019–August 2, 2019, (Stroudsburg, PA: Association for Computational Linguistics (ACL)), 4710–4723.

Nian, M., Dong-Yuan, S., and Xian-Zhong, D. J. P. S. T. (2008). A novel method for fast power network topology formation and partial revision based on graph theory. *Power Syst. Technol.* 32 (13), 35–39.

Nikolentzos, G., Dasoulas, G., and Vazirgiannis, M. (2020). k-hop graph neural networks. *Neural Networks* 130, 195–205. doi:10.1016/j.neunet.2020.07.008

Park, N., Kan, A., Dong, X. L., Zhao, T., and Faloutsos, C. (2019). Estimating node importance in knowledge graphs using graph neural networks. Proceedings of the 25th Acm Sigkdd International Conferencce on Knowledge Discovery and Data Mining Kdd'19, Anchorage, AK, August 4–8, 2019, (New York, NY: ACM), 596–606.

Song, S.-Q., Zhu, Y.-L., and Yu, H. (2005). A power network topology tracking method based on graph theory and artificial intelligence search technique. *Power Syst. Technol.* 29 (19), 45–49.

Su, C., Gao, S., and Li, S. (2020). GATE: graph-attention augmented temporal neural network for medication recommendation. *IEEE Access* 8, 125447–125458. doi:10.1109/Access.2020.3007835

Wang, S., Xu, Q., Zhang, G., and Yu, L. (2009). Modeling of wind speed uncertainty and interval power flow analysis for wind farms. *Dianli Xitong Zidonghua/Autom. Electr. Power Syst.* 33 (21), 82–86. doi:10.1016/j.apm.2007.10.019

Wang, T., Liu, Y., Gu, X., and Qin, X. (2019). Vulnerable lines identification of power grid based on cascading fault space-time graph. *Zhongguo Dianji Gongcheng Xuebao/Proc. Chin. Soc. Electr. Eng.* 39 (20), 5962–5972. doi:10.13334/j.0258-8013.pcsee.181730

Wu, B., Liu, Y., Lang, B., and Huang, L. (2018). DGCNN: disordered graph convolutional neural network based on the Gaussian mixture model. *Neurocomputing* 321, 346–356. doi:10.1016/j.neucom.2018.09.008

Wu, Z., Pi, D., Chen, J., Xie, M., and Cao, J. (2020). Rumor detection based on propagation graph neural network with attention mechanism. *Expert Syst. Appl.* 158, 113595. doi:10.1016/j.eswa.2020.113595

Xie, Y., Xu, H., Li, J., Yang, C., and Gao, K. (2020). Heterogeneous graph neural networks for noisy few-shot relation classification. *Knowl. Base Syst.* 194, 105548. doi:10.1016/j.knosys.2020.105548

Yansheng, L., Jing, L. I., Yadi, L., Lingyun, W. U., Qiang, L. I., Jun, Z., et al. (2017). Large power grid topology analysis based on graph partitioning. *Power Syst. Prot. Control* 45 (23), 108–115. doi:10.7667/PSPC171060

Yu, M., Zhang, Z., Li, X., Yu, J., Gao, J., Liu, Z., et al. (2020). Superposition graph neural network for offshore wind power prediction. *Future Generat. Comput. Syst.* 113, 145–157. doi:10.1016/j.future.2020.06.024

Zhang, C., Yu, J. J. Q., and Liu, Y. (2019). Spatial-temporal graph attention networks: a deep learning approach for traffic forecasting. *IEEE Access* 7, 166246–166256. doi:10.1109/Access.2019.2953888

Zhu, W., Zhou, X., and Tang, R. (2011). Elasticity network topology mapping for power grid. *Zhongguo Dianji Gongcheng Xuebao/Proc. Chin. Soc. Electr. Eng.* 31 (31), 109–117. doi:10.1016/S1872-5805(11)60064-4

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.