Modeling of Fault Propagation Paths in Power Communication Networks Combined with GNN

Lina Hu, Yi Tang, Lu Ye
State Grid Hubei Information & Telecommunication Company, Wuhan 430048, Hubei, China
Corresponding author: Yi Tang (e-mail: hbxttangyi@163.com)

ABSTRACT Power communication network faults often propagate with changes in node status and network topology, and their significant dynamic evolution characteristics are difficult to capture effectively. Capturing these dynamically evolving faults is critical to guaranteeing the reliability of power communication links and continuous industrial operation. To this end, this paper proposes a method combining graph neural networks (GNNs) to construct a propagation path modeling framework that integrates topological structure and node status time series characteristics. A time-slice network snapshot is constructed based on the node status and communication link. The temporal feature extraction is enhanced by position encoding and sliding time window. The T-GCN (Temporal Graph Convolutional Network) model is combined with the graph topology and dynamic propagation mode to predict the failure probability of each node in the power communication network and generate a propagation path diagram. Finally, the attention mechanism is used to identify key nodes, and the actual propagation trajectory is used to supervise the training to achieve accurate early warning of potential paths. Experiments show that the proposed model has a precision of 0.92 in node fault prediction, an average time error of only 1.2 minutes, a key node coverage of 90.5%, a propagation delay reduction of 31.2%, an average coverage of 93.2%, and a minimum F1-score of 0.84 in diverse fault scenarios, demonstrating high precision and strong generalization ability. Because the method models node state, topology and propagation delay, it is directly applicable to reliability analysis of electromagnetic communication links in power networks.

INDEX TERMS power communication network, fault propagation, graph neural network, temporal graph model, electromagnetic communication links

I. INTRODUCTION

THE power communication network serves as the foundational infrastructure that supports the operation, dispatching, protection, and control of modern power systems. This essential communication backbone ensures the reliable and high-quality energy supply crucial for continuous industrial processes, particularly in highly automated and energy-intensive sectors. With the increasing level of informationization and automation of power systems, its stability and fault resistance have a decisive impact on the overall safe operation of the power system. In the power communication network, data is exchanged between communication nodes through links. The failure of any node or link may cause system-level cascading risks [1,2]. Therefore, accurately predicting the propagation path of faults in the communication network in a complex power grid environment has important theoretical value and practical significance. The power communication network has the characteristics of complex topology and strong heterogeneity. At the same time, its operating state evolves over time. The changes in node and link status have a direct impact on the fault propagation path [3,4].

Most current related studies still model the communication network based on static topology, only analyzing the network status at a certain moment, ignoring the time evolution characteristics of node attributes. This static modeling method is difficult to truly characterize the propagation dynamics of communication faults [5,6]. At present, the research on fault propagation in power communication networks mainly focuses on two directions. On the one hand, the propagation path is modeled and analyzed through graph theory methods. This type of research is often based on a fixed topological structure and uses static analysis to infer the propagation trend, lacking effective modeling of time factors [7,8]. On the other hand, some studies have attempted to introduce propagation probability models or propagation rules to simulate the fault diffusion process, such as Markov chain models, Petri nets, neural networks, etc., but these methods generally rely on artificially set propagation logic or fixed fault diffusion rules, and are difficult to adapt to the actual propagation behavior in complex networks [9,10]. Although existing studies have achieved certain results in path visualization and key node identification, they generally have problems such as insufficient generalization ability and poor dynamic adaptability.

In response to the above problems, some scholars have begun to try to introduce data-driven methods to model fault propagation paths in power communication networks in recent years [11]. Models represented by deep learning have gradually attracted attention due to their excellent feature extraction and nonlinear modeling capabilities. In this context, GNN, as a deep learning model that can process non-Euclidean structured data, has the ability to learn dependencies between nodes from graph structures [12,13], providing a new approach to solving fault propagation modeling in power communication networks. Some studies have used static GNN models to identify key communication nodes or links, extracted node features through multi-layer graph convolution operations, and used this as the basis for propagation path prediction [14,15]. However, static GNN still has difficulty in capturing the temporal variation of node states and lacks the ability to characterize the temporal evolution of power fault propagation paths [16,17].

This study conducts system design in terms of structural modeling, time series modeling, and path generation, and constructs a dynamic fault propagation prediction framework for power communication networks. The model performs well in the node fault prediction task, with a precision of up to 0.92 and an average prediction time error of less than 1.2 minutes, showing excellent timeliness. At the same time, its key node coverage reaches 90.5%; the average coverage is 93.2%; the propagation delay is significantly reduced by 31.2%. When dealing with various types of faults, the F1-score is as low as 0.84 in a variety of fault scenarios, which is better than the GCN-LSTM, GAT, and GNN models, indicating that the model has strong accuracy and generalization capabilities, and is more stable in adapting to diverse and complex network environments. The modeling method proposed in this study can significantly improve the accuracy of propagation path prediction and enhance the model's adaptability to various fault behaviors. This advance in intelligent fault warning auxiliary tools, characterized by its scalability and deployability, directly aids power system operation and maintenance departments in guaranteeing the stable, high-quality energy supply that is critical for maintaining the highspeed and precision operations of energy-intensive sectors.

II. DYNAMIC PROPAGATION MODELING MECHANISM OF POWER COMMUNICATION NETWORK

A. CONSTRUCTION OF DYNAMIC POWER COMMUNICATION GRAPH

1) Graph Structure Generation Mechanism

The structural data of the power communication network comes from the link topology log and historical alarm information database in the communication network management system. To realize the graph structure input of propagation modeling, it is necessary to map the original data into a graph, where node $v_i \in V$ represents a communication device in the network, such as a router, switch, protection device terminal, etc.; edge $e_{ij} \in E$ represents the actual communication link between nodes v_i and v_j, which is generated based on the physical connection relationship of the optical cable or the logical tunnel mapping result. All edges are directional, and a directed graph $G = (V, E)$ is used to represent the network status to more accurately describe the information propagation path in the network [18].

The node attribute vector is constructed using sparse coding, with an initial dimension of 12. The fields include node type (terminal/aggregation device/core device), device manufacturer code, operating status label (normal/warning/fault), device load, physical temperature, whether it is a redundant backup mark, etc. Four features are introduced into the edge attributes: link bandwidth, link delay, link stability score, and real-time link utilization, as the attribute matrix input of the edge. The above node and edge feature values are uniformly normalized and represented as node feature matrix $X_t \in \mathbb{R}^{n\times d}$ and edge feature matrix $E_t \in \mathbb{R}^{m\times p}$, respectively.

For some missing link state data, the historical average filling and similar device alignment strategy are used to patch them to ensure that the dynamic graph sequence is continuous in the time dimension. All graph structures are converted into adjacency matrix form and saved. Let the adjacency matrix be $A_t \in \{0, 1\}^{n\times n}$, which is 1 if the edge exists and 0 otherwise, forming the structural input part of GNN [19].

2) Time Slice Division and Graph Snapshot Management

The construction of dynamic graphs requires extracting network status snapshots at different time points within a continuous time range, and on this basis, forming a graph sequence $\{G_t\}_{t=1}^{T}$. This study sets the time granularity to 5 minutes, extracts the communication status data throughout the day according to a fixed window sliding, and generates a total of 288 graph snapshots, corresponding to the power communication network status of each time slice. This granularity refers to the fault response time configuration of a provincial power communication master station, which can cover the common link fluctuations and equipment status change range, ensuring that the modeling is sufficiently timely. Each graph $G_t = (V_t, E_t, X_t, E_t)$ in each time slice t is based on the equipment status and link attributes at that moment to build a consistent node set, and the edge set is allowed to change to reflect dynamic behaviors such as link interruption and automatic switching. To more accurately capture the changes in node attributes over time, the sliding window technology is used to extract the node state sequence. This method can smooth the mutation problem caused by the change of node state and provide time context for subsequent model learning.

Figure 1 shows the dynamic topology evolution of a typical substation power communication network at different times. The node index is the horizontal and vertical coordinates, and the color depth represents the existence of a link between nodes. Blue indicates that a link exists, and white indicates no connection. The topology at the initial moment (t=1) is relatively dense, reflecting the stable communication connection between the core node and the edge node, especially the high connection density of the core node group, which is in line with the hierarchical characteristics of the power communication network. In the middle period (t=5), the topology fluctuates significantly, and the edge node links are randomly disconnected and newly created, reflecting the network's flexible adjustment ability in a dynamic environment. The final state (t=10) shows that the network as a whole tends to be stable, but some dynamic changes are still retained to ensure the reliability and connectivity of the network.

B. EXTRACTION OF NODE STATE TIME SERIES FEATURES

1) Sliding Time Window Mechanism

To model the evolution of node state over time, this paper uses a sliding time window mechanism to extract the time series evolution features of nodes and construct a node-level multidimensional time series. In the specific operation, a fixed time window length $w = 6$ is set, corresponding to the node state changes in the last 60 minutes (sampling interval is 10 minutes), and the node features in each time window are spliced into a tensor input sequence. For any node v_i, its state sequence at time t is expressed as Formula 1:

$$S_i^{(t)} = \{x_i^{(t-w+1)}, \ldots, x_i^{(t)}\} \in \mathbb{R}^{w \times d} \tag{1}$$

In formula 1, $x_i^{(t)} \in \mathbb{R}^d$ is the feature vector of the node at time t, and the dimension $d = 16$ includes historical evolution features such as load rate, device temperature, voltage offset rate, and link score.

After the time dimension is constructed, the gated recurrent unit (GRU) is used to encode the node sequence. Assuming that the hidden dimension of the GRU unit is 64, the weights are shared; the nodes are encoded independently; the time series representation $h_i^{(t)} \in \mathbb{R}^{64}$ of each node at the current time is output. This encoding method not only captures the trend of node attribute changes, but also introduces a time memory mechanism to effectively solve the problem of feature jumps caused by drastic fluctuations in node attributes in a short period of time. To ensure that each sample contains enough historical information, only samples with sliding window lengths that meet the condition $w \leq t$ are selected for modeling during the training phase. Figure 2 shows the state evolution characteristics of key nodes in the power communication network in a specific period of time, focusing on the dynamic changes of timing characteristics and their correlation with the fault time. In the left figure, the X-axis represents continuous time steps (10 minutes per unit, covering 8 hours), and the Y-axis represents the values of three key indicators, including load rate (Load), temperature (Temp), and voltage (Voltage). Between the 25th and 28th time steps, all three indicators fluctuated significantly, especially the load and temperature reached their peaks at the 26th time step, indicating that an abnormal event may have occurred during this period. The right figure further depicts the change in similarity between the node encoding features at different time steps and the fault start time (step 25). The X-axis also represents the time step, and the Y-axis represents the cosine similarity. The red mark highlights the significant increase in similarity during the fault, indicating that the disturbance caused by the fault is clearly distinguishable in the feature space. This result verifies that the feature expression after the introduction of temporal position encoding has sensitivity and distinguishing ability.

2) Positional Encoding Embedding

While recurrent structures like GRU can model time dependencies, their perception of the relative position of time steps is implicit, making it difficult to distinguish the different contributions of "recent mutations" and "historical cumulative effects" to fault prediction. Therefore, this paper introduces explicit positional encoding in the time dimension to enhance the model's sensitivity to fault evolution stages. Specifically, this paper employs a fixed sine-cosine positional encoding consistent with Transformer, generating a position vector p_i of dimension d for each time step $i \in \{1, \ldots, L\}$ within the sliding window. The k-th dimension is defined as follows: if k is even, then:

$$p_{i,k} = \sin\left(\frac{i}{10000^{k/d}}\right) \tag{2}$$

If it is an odd number, then:

$$p_{i,k} = \cos\left(\frac{i}{10000^{(k-1)/d}}\right) \tag{3}$$

This encoding is added element-wise to the original node feature x_i^{t-k}, forming a position-enhanced input: $\tilde{x}_i^{t-k} = x_i^{t-k} + p_k$.

This design allows the model to explicitly distinguish between state changes "closer to the current moment" and "earlier moments" within a time window, thereby more accurately identifying temporal patterns of fault precursors. It is important to note that "position" here refers to the relative time step sequence within the time series, rather than the spatial coordinates shown in the diagram. Its role is not to replace the temporal modeling of the GRU, but rather to provide it with structured temporal priors, improving feature discriminative power.

C. TIME-AWARE GNN CONSTRUCTION

This study employs a Temporal Graph Convolutional Network (T-GCN) as its basic architecture due to its good

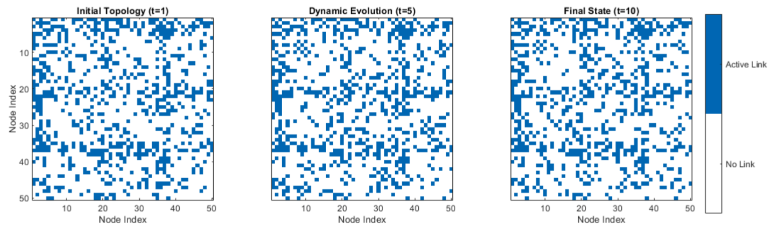

FIGURE 1. Dynamic topology evolution of a typical substation power communication network at different times

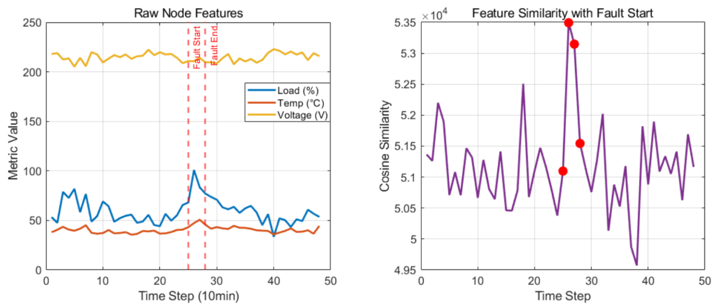

FIGURE 2. State evolution characteristics of key nodes in the power communication network in a specific period of time

balance between jointly modeling graph structure dependencies and temporal dynamics. The T-GCN is composed of a cascaded Graph Convolutional Network (GCN) and Gated Recurrent Units (GRUs). The GCN is responsible for aggregating the spatial information of neighboring nodes at each time step, while the GRU models the evolution of node states along the time dimension. The input data is a dynamic graph sequence $\{(A_t, X_t)\}_{t=1}^{T}$, where $A_t \in \{0,1\}^{N\times N}$ is the adjacency matrix at time t, representing the physical communication links between power communication devices (including optical cable connections and logical tunnel mappings), and $X_t \in \mathbb{R}^{N\times d}$ is the node state time series, with each row corresponding to a node's multidimensional feature vector at that time, including 12 normalized indicators such as device type, load rate, temperature, voltage offset rate, and operating status label. To enhance the model's ability to express complex fault propagation patterns, a location-aware temporal embedding mechanism and an attention-guided training strategy under propagation path constraints on the basis of the standard T-GCN are introduced, enabling it not only to capture the local neighborhood fault propagation trend, but also to identify global key propagation hubs.

Despite the existence of various spatiotemporal graph neural network architectures (such as DCRNN, ASTGCN, STGCN, etc.), this paper chooses T-GCN mainly based on three considerations: First, T-GCN adopts a cascaded structure of GCN+GRU, which has high computational efficiency and a moderate number of parameters, making it suitable for deployment in provincial power communication networks with a node scale of up to 2000. Second, its non-cyclic spatiotemporal coupling method (spatial aggregation followed by temporal recursion) is more in line with the local propagation characteristics of power communication faults—faults usually propagate between neighboring devices first and then evolve over time. Finally, compared with methods such as ASTGCN that rely on predefined spatial attention or graph diffusion convolution, T-GCN is more adaptable to dynamic topology changes because it uses the current adjacency matrix A_t for graph convolution at each time step, which can directly reflect real-time topology disturbances such as link interruption or switching. To realize the spatiotemporal dynamic modeling of fault propagation in power communication networks, this paper adopts the time-aware GNN model, namely Temporal Graph Convolutional Network (T-GCN). Figure 3 shows the complete architecture of a fault propagation path prediction model based on dynamic graphs, covering key links such as input, feature extraction, fusion, output, and path generation. The input part is time-series dynamic graph data, including node feature matrix and adjacency matrix. Spatial features are extracted by two layers of graph convolution, and temporal features are extracted by two layers of GRU, generating spatial feature matrix and temporal feature vector, respectively. The fusion module concatenates the two types of features and outputs the fault probability of each node through the fully connected layer and the Softmax function. The output module generates the fault propagation path within the next three steps through threshold screening and topological constraint mechanism, and identifies the key nodes. The model is trained with the Adam optimizer, introduces cross entropy loss and Dropout regularization, and has good spatiotemporal modeling capabilities and prediction effects.

The model combines the advantages of the Graph Con-

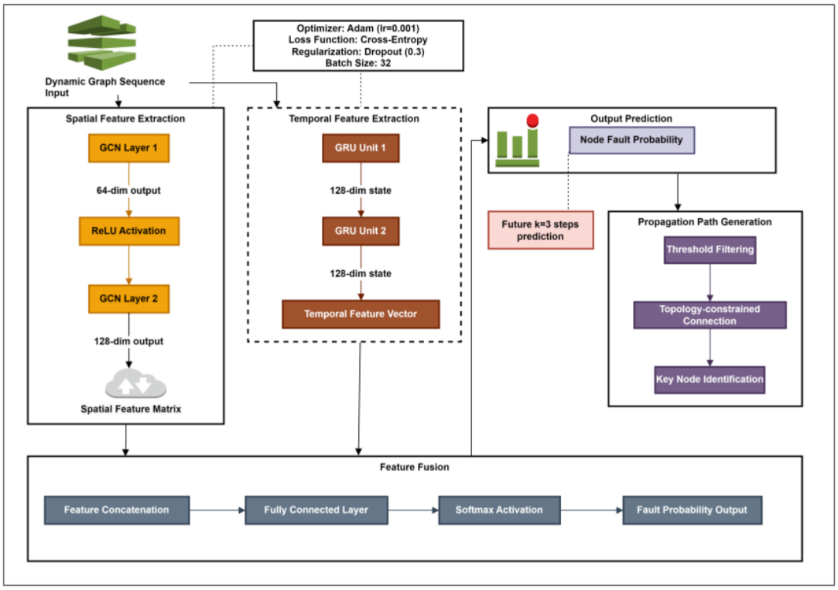

FIGURE 3. Model framework diagram of fault propagation path prediction model based on T-GCN

volutional Network (GCN) and the Gate Recurrent Unit (GRU), which are responsible for spatial structure information extraction and time series feature modeling respectively. It can effectively capture the complex spatial dependencies between nodes and the time evolution characteristics of their states, and then learn the potential patterns of fault propagation [20]. Specifically, the basic data of the input model is the dynamic graph sequence $\{G_t\}_{t=1}^{T}$ constructed in the previous chapter. Each graph snapshot contains the node feature matrix $X_t \in \mathbb{R}^{N\times d}$ and the adjacency matrix $A_t \in \mathbb{R}^{N\times N}$, where N is the number of network nodes and d is the node feature dimension. The model first performs a graph convolution operation on the graph snapshot of each time step to capture the local correlation features of the nodes in the current topological structure. The graph convolution uses a normalized Laplacian matrix, as shown in Formula 4:

$$\tilde{L}_t = I - D_t^{-\frac{1}{2}} A_t D_t^{-\frac{1}{2}} \tag{4}$$

Formula 4 is used as the convolution kernel to ensure that the feature aggregation process is stable and effective. The convolution layer parameters are set to a two-layer GCN structure with hidden layer dimensions of 64 and 128, respectively. After completing the spatial convolution, the spatial feature sequence of the node is sent to the timing module. The timing module uses a two-layer GRU network with the number of hidden layer units in each layer set to 128, and uses its gating mechanism to solve the long-term and short-term dependency problems. GRU can effectively integrate the historical state of node features and capture the dynamic evolution trend of fault status . In the specific calculation process, the input of each time step is the node feature vector output by the graph convolution at the corresponding time. After passing through GRU, the time-aware hidden state vector h_t is obtained, whose dimension is 128.

The overall structure of the model jointly optimizes the spatial convolution kernel and the time recursive parameters through end-to-end training. The training goal is to minimize the cross entropy loss function of multi-step fault prediction. The supervision signal comes from the time series of the real fault label to ensure that the model learns the spatiotemporal dependency of node fault propagation. The Adam optimizer is used in the training process; the initial learning rate is set to 0.001; the batch size is 32; the maximum number of training rounds is 100. To prevent overfitting, the Dropout mechanism is introduced with a ratio of 0.3, which is applied between the graph convolution layer and the GRU layer.

Figure 4 shows the training process and feature space evolution of the T-GCN model in the power communication network node fault detection task, reflecting the changing rules of model performance optimization and semantic feature clustering. In the left figure, the X-axis is the number of training rounds (Epoch); the Yaxis is the indicator value; the three curves represent the training loss, validation loss, and F1 score, respectively. It can be seen that as the number of training rounds increases, the loss function decreases rapidly and converges, eventually approaching 0.1. The F1 score continues to rise and finally stabilizes at around 1.0, indicating that the model has good fitting ability and generalization performance in fault identification tasks. The middle and right figures use the t-SNE (t-Distributed Stochastic Neighbor Embedding) method to reduce the dimension of the potential features of the nodes at different training times. The X-axis and Y-axis represent the two principal components of t-SNE. From t=1 to t=5, it can be clearly observed that the node feature distribution gradually evolves from the initial mixed state to a clear cluster structure, indicating that T-GCN has effectively extracted discriminative spatiotemporal features through multiple rounds of training, and can more accurately distinguish potential faults from normal nodes.

The T-GCN model achieves spatiotemporal joint encoding of node states by deeply fusing spatial graph convolution with temporal recursive units. This method overcomes the defect of traditional static graph models that ignore time evolution, and also avoids the deficiency of pure time series models that cannot model complex topological structures.

D. FAULT PROPAGATION PATH LEARNING MECHANISM

1) Propagation Path Graph Generation

Based on the output results of the T-GCN model, this paper designs a learning mechanism for fault propagation paths to achieve quantitative prediction of future node failure risks and identification of key propagation paths . The model outputs the failure risk probability of all nodes in the corresponding network at each prediction time step, recorded as matrix $P_t \in \mathbb{R}^N$, where N represents the total number of nodes, and $P_{t,i}$ represents the failure probability of node i at time t. To construct the fault propagation path diagram, a threshold screening method is used.

The probability threshold θ is set (usually 0.5 as the initial value, adjusted according to the specific network situation), and all nodes are regarded with risk probability greater than θ as potential fault nodes. The edges between nodes are connected through the original network topology constraints and propagation rules to form a propagation path diagram, as shown in Formulas 5 and 6:

$$G_t^p = (V_t^p, E_t^p) \tag{5}$$

$$V_t^p = \{v_i \mid P_{t,i} > \theta\} \tag{6}$$

The edge set E_t^p is determined by the communication link and the propagation association in the time series.

2) Identification of Key Propagation Nodes

To improve the interpretability and prediction accuracy of fault propagation modeling, this paper deeply integrates the graph attention mechanism into the propagation path inference process, rather than just using it for post-processing analysis. Specifically, a multi-head graph attention network (GAT) layer is constructed on the high-risk node subgraph output by T-GCN. This layer adaptively assigns attention weights to the neighbors of each node during the message passing phase. The weight calculation is based on a learnable linear transformation between the node's hidden state and its neighbor features, and is normalized using LeakyReLU activation and softmax. This mechanism enables the model to dynamically focus on the neighbor nodes with the greatest influence on propagation when predicting the fault probability at the next time step, thereby enhancing the ability to identify key nodes. The attention weights are not only used to generate the propagation path graph, but also participate in joint training as auxiliary supervision signals: nodes on the actual propagation path in the historical fault trajectory are used as positive samples to guide the attention distribution to concentrate on real key nodes. The multi-head is set to 8 heads; the hidden dimension is 128; the regularization coefficient is 0.01 to balance concentration and diversity.

The left sub-figure of Figure 5 shows the statistical distribution of node failure probability. The horizontal axis represents the failure probability value, and the vertical axis represents the number of nodes falling within the probability interval. The failure probability of most nodes is concentrated between 0.2 and 0.5, and the probability distribution shows a certain skewness. At the same time, the red dotted line marks the fault judgment threshold θ=0.5, which is used to distinguish potential fault nodes from non-fault nodes. In the simulation environment, the number of nodes exceeding this threshold is relatively small, reflecting the model's ability to sparsely identify abnormal nodes. The right subgraph plots the weights assigned to the first ten nodes by the attention mechanism in the model. The horizontal axis is the node number, and the vertical axis is the attention weight value. The higher the value, the greater the importance of the node in the reasoning process. Node numbered 4 receives the highest weight α=0.25, indicating its critical role in the fault propagation path.

During the training process, attention weight and fault probability are used as supervision targets together, and the actual fault propagation trajectory is used as a label to optimize the model parameters through a joint loss function. The loss function includes cross entropy loss to improve the accuracy of node fault prediction, and attention weight regularization to guide the model to focus on the propagation effect of key nodes. The regularization parameter is set to 0.01 to prevent excessive concentration or diffusion of attention distribution and maintain reasonable propagation interpretation ability.

E. DESIGN OF ABNORMAL PROPAGATION PREDICTION MODULE

1) Design of Supervision Signal and Loss Function

The supervision signals relied upon for model training originate from the spatiotemporal sequences of real fault events recorded in historical operation logs, rather than complete, predictable future trajectories. Specifically, 217 independent fault events are extracted from the alarm logs of a provincial power communication network over six consecutive months. Each event is labeled with the fault initiation node, secondary fault nodes along the propagation path, and their corresponding timestamps (accurate to 10 minutes). These labeled historical propagation segments constitute the training samples. The model's task is to predict the fault probability of each node in multiple future time steps, given a normal/abnormal state sequence from the previous 60 minutes. Therefore, the model does not

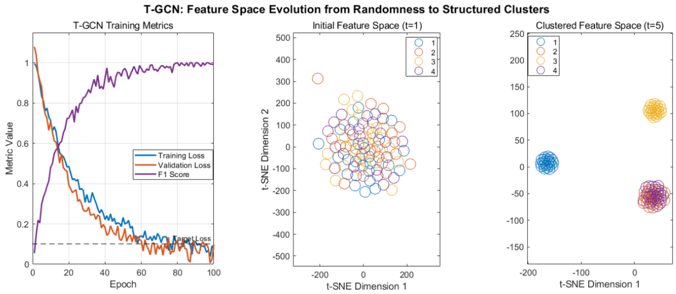

FIGURE 4. Model training process and feature space evolution

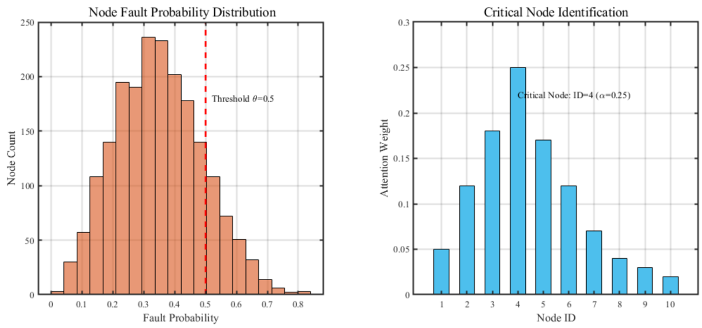

FIGURE 5. Statistical distribution of node failure probability and node assigned weights

perform interpolation under a "known complete path," but rather performs forward prediction within a limited historical window. Its generalization ability relies on inductive learning of the spatiotemporal patterns of fault propagation. This design ensures that the model can operate with only real-time monitoring data in actual deployment, without requiring future information. To optimize model parameters, a cross-entropy loss function for multi-step sequence classification is used. This loss function is calculated for each node's binary classification task (fault/non-fault), as shown in Formula 7:

$$L = -\frac{1}{N \times T} \sum_{i=1}^{N} \sum_{t=1}^{T} [y_{i,t} \log \hat{y}_{i,t} + (1 - y_{i,t}) \log(1 - \hat{y}_{i,t})] \quad (7)$$

In Formula 7, N is the number of nodes; T is the number of prediction time steps; $y_{i,t}$ is the node's true state label; $\hat{y}_{i,t}$ is the fault probability output by the model. The cross entropy loss function fully considers the prediction accuracy of multiple time steps, effectively guides the model to learn the time dynamic change characteristics of the node fault state, and reduces the false alarm and missed alarm rate.

2) Path Reasoning and Anomaly Identification in the Prediction Stage

In the prediction stage, the module automatically infers the node failure risk probability at future moments based on the model weights obtained through training. Combined with the threshold judgment mechanism, it screens potential secondary failure nodes and identifies the communication links and areas that may be affected. The core of this stage is to convert the temporal probability results output by the model into a continuous propagation path to form a spatiotemporal diffusion map of future faults.

The upper sub-figure in Figure 6 shows the sensitivity changes of model performance indicators under different probability thresholds. The horizontal axis is the probability threshold θ of fault judgment, and the vertical axis is the corresponding precision, recall, and F1 score. As the threshold increases from 0.1 to 0.9, F1 gradually increases and approaches 0.95 when the threshold is close to 0.75. This shows that in this model, the threshold setting has a significant impact on the fault identification results, and a reasonable threshold selection is particularly critical for balancing false positives and false negatives. The sub-figure below depicts the changes in the number of real and predicted fault nodes in the time series, with the horizontal axis being the time step and the vertical axis being the number of fault nodes at each moment. The actual curve fluctuates periodically. In 20 time steps, the model prediction results can basically follow the real trend, indicating that the model has strong propagation coverage and timing response effects.

III. EVALUATION SYSTEM FOR FAULT PROPAGATION PREDICTION

To comprehensively evaluate the model's prediction performance for the propagation path of power communication network faults, this study adopts a multi-dimensional evaluation index system. The data set used in the evaluation experiment is based on the actual operation log of a provincial power communication network, covering 6 consecutive

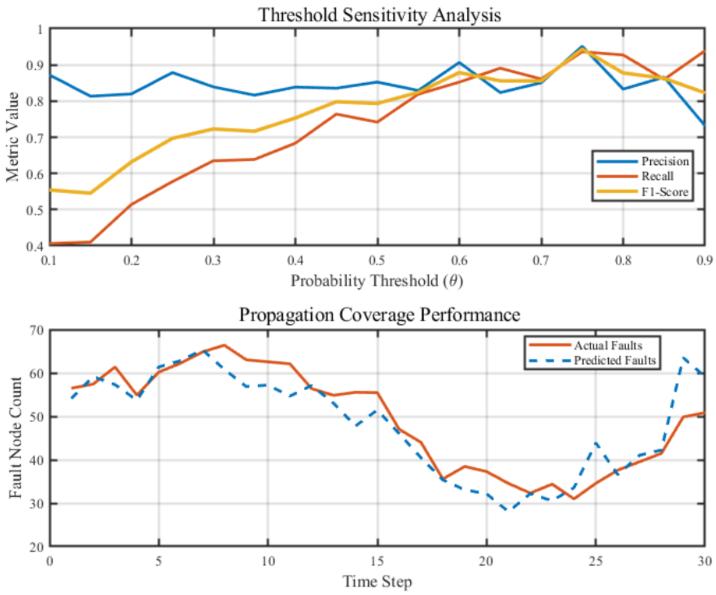

FIGURE 6. Threshold sensitivity analysis and propagation path coverage effect

months of network status and fault event records with a time resolution of 10 minutes. The data set includes node status, link attributes, and alarm information, covering a variety of fault types and propagation modes, and can truly reflect the dynamic evolution of the network under different load and environmental conditions. Through strict data cleaning and structured processing, a dynamic graph sequence with labels is constructed to ensure the temporal continuity and spatial consistency of the training set and the validation set during the evaluation process. The dataset includes about 2,000 nodes and 5,000 links, which is highly complex and representative.

A. NODE PREDICTION AND RECOGNITION EFFECT

The node prediction accuracy evaluation focuses on the performance of the model in identifying faulty nodes. By comparing the set of faulty nodes predicted by the model with the set of actual faulty nodes, the three indicators of Precision, Recall, and F1-score are calculated. Precision reflects the proportion of nodes predicted to be faulty that are actually correct; recall measures the proportion of actual faulty nodes that are correctly predicted; F1-score, as the harmonic average of the two, comprehensively reflects the overall recognition effect of the model. During the evaluation process, the predicted nodes are first compared with the actual nodes according to the time step; the number of true positives, false positives, and false negatives is counted; then the three indicators are calculated. This indicator group can fully reflect the model's ability to locate faulty nodes, and is particularly suitable for abnormal node detection tasks.

The left side of Figure 7 shows the performance comparison of five models in the fault propagation node identification task. The X-axis is the model name, and the Y-axis is the performance score of the three indicators Precision, Recall, and F1-score. The proposed model is significantly better than the single T-GCN model, GCN-LSTM (Graph Convolutional Network + Long Short-Term Memory), GAT and GNN models in all three indicators, with Precision as high as 0.92 and F1-score as high as 0.905, indicating that this method has significant advantages in recognition accuracy and stability. The right side of Figure 7 shows the recognition effect of different types of faults under this model, with the X-axis being F1-score and the Y-axis being fault category. The recognition effect is generally good among the 10 common faults. Among them, Hardware Fault, Cable Cut, and Link Failure all exceed 0.91, and the lowest is maintained above 0.84. The F1-score of EMI (Electromagnetic Interference) is 0.88. Overall, the model has strong adaptability in diverse scenarios and has good generalization performance and practical application value.

B. PATH OVERLAP RATE EVALUATION

The path overlap rate is used to quantify the similarity between the predicted propagation path and the actual propagation path. The specific steps are to represent the fault propagation path as a node sequence, and evaluate the overlap between the model predicted path and the actual path by node matching. When calculating, this paper first compares the node sequences in the two paths, and counts the number of occurrences of the same nodes and their order relationship. A high overlap rate indicates that the predicted path reproduces the actual propagation trajectory well, reflecting the model's ability to capture the spatiotemporal structure of the propagation path. This indicator pays special attention to the integrity and continuity of the path, and is one of the core indicators for evaluating the accuracy of

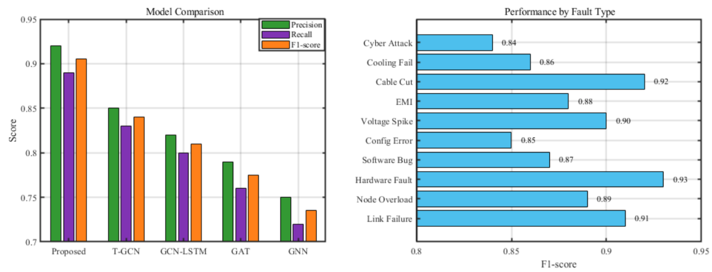

FIGURE 7. Multi-model performance and fault type identification effect

propagation path modeling. Figure 8 shows the change trend of path overlap rate of different models under various path lengths. The horizontal axis is the path length (number of nodes), and the vertical axis is the path overlap rate, which reflects the ability of each model to restore the real path under different network scales. As the path length increases, the overlap rate of most models shows a downward trend, indicating that the model's ability to maintain the global structure gradually weakens under long path conditions. Among them, the model in this paper always maintains a high overlap rate at all path lengths, especially when the path length is 20, it is still close to 0.85, and it is significantly better than the comparison methods such as GNN and GAT at other path lengths, reflecting its stability and accuracy in the reconstruction of complex network structures.

C. DYNAMIC EVOLUTION MATCHING ERROR

The dynamic evolution matching error measures the accuracy of the model's dynamic prediction of fault propagation time. By setting a time window, the time point when the actual fault node occurs is compared with the time point predicted by the model, and the average time deviation is calculated. This evaluation examines whether the node is correctly predicted and also focuses on the time consistency of the node fault state. A smaller time deviation means that the model can more accurately capture the temporal characteristics of fault propagation, which helps to improve the timeliness of early warning. The dynamic evolution matching error evaluates the prediction ability of the model in the time dimension, making up for the deficiency of the simple spatial matching index. Table 1 shows the performance of the five models in fault propagation time prediction, specifically comparing and analyzing the average time error, maximum time error, time consistency, and main error sources. The model proposed in this paper has an average time error of only 1.2 minutes, a maximum time error of 3.5 minutes, and a time consistency of up to 92.8%, which is significantly better than other comparison models, reflecting its accuracy and stability in time prediction. In contrast, the average time error of the GNN model reaches 3.2 minutes; the maximum error is 7.5 minutes; the time consistency is only 79.8%, indicating that its prediction performance is poor under dynamic topological changes. Although the T-GCN and GAT models perform second best in terms of time error, their time consistency is 89.6% and 87.5%, respectively, indicating that they have a good grasp of the temporal characteristics of the propagation process to a certain extent. In terms of the main sources of error, the errors of the model in this paper mainly come from sudden failures, which reflects its sensitivity to complex abnormal events.

TABLE 1. Performance of five models in fault propagation time prediction

Model	Avg. Time Error (min)	Max. Time Error (min)	Temporal Consistency (%)	Primary Error Source
Proposed	1.2	3.5	92.8	Sudden faults
T-GCN	1.5	4.2	89.6	Propagation acceleration
GCN-LSTM	2.8	6.1	84.3	State transitions
GAT	2	5.3	87.5	Neighbor interference
GNN	3.2	7.5	79.8	Topological changes

D. EVALUATION OF THE INFLUENCE OF KEY PROPAGATION NODES

This indicator evaluates the influence contribution of the key propagation nodes identified by the model in the actual fault propagation process. First, the key nodes are ranked based on the weights output by the attention mechanism, and the top-ranked nodes are selected as the key node set. Then, by analyzing the coverage of these nodes in the actual fault propagation path and their control over the propagation chain, the driving ability of the key nodes on fault propagation is quantified. The specific evaluation steps include calculating the position ratio of key nodes in the actual propagation path and their impact on the path propagation length and speed. The influence of key propagation nodes reflects the effect of the model in capturing the core hub nodes of propagation, which helps to judge the model's ability to understand and explain the fault propagation mechanism.

Table 2 shows the comparison of multiple key performance indicators of the five models in fault propagation analysis, including coverage, impact factor, propagation control capability, key node sorting accuracy, and propagation delay reduction rate. The model proposed in this paper performs well in all indicators, with a coverage rate of 90.5%, which is significantly higher than the second-best T-GCN (87.3%) and other models, indicating that it can capture the fault propagation path more comprehensively. The impact factor and propagation control capability are 0.95 and 0.92, respectively, which shows that the model

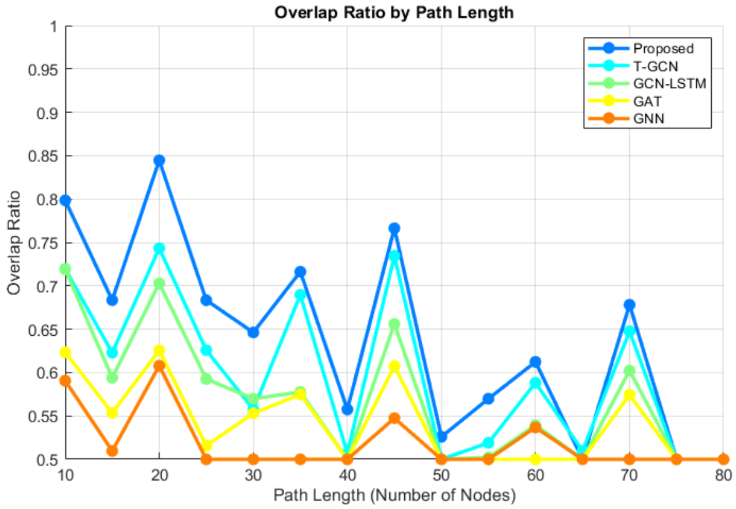

FIGURE 8. Changes in path overlap rate of different models under various path lengths

has strong capabilities in assessing the impact range of faults and effectively controlling propagation. The key node sorting accuracy is as high as 94.2%, which is better than T-GCN's 90.5%.

TABLE 2. Comparison of multiple key performance indicators in fault propagation analysis

Evaluation Metrics	Proposed	T-GCN	GCN-LSTM	GAT	GNN
Coverage (%)	90.5	87.3	82.1	85.6	78.9
Impact Factor	0.95	0.89	0.81	0.87	0.76
Propagation Control Capability	0.92	0.86	0.79	0.84	0.73
Criticality Ranking Accuracy (%)	94.2	90.5	85.7	88.9	81.3
Propagation Delay Reduction (%)	31.2	26.8	19.4	23.7	16.5

E. PROPAGATION PATH COVERAGE

The propagation path coverage is used to evaluate the coverage of the predicted path to the actual fault propagation range. The calculation method is to calculate the ratio of the number of nodes contained in the predicted path to the total number of nodes in the actual propagation path. The higher the coverage, the more comprehensive the model can predict the propagation range and reduce omissions. This indicator helps evaluate the model's coverage of fault diffusion in the spatial dimension. It is particularly important for timely warning of secondary faults in power communication networks. It can assist in judging the applicability and generalization ability of the model in different network topologies and fault modes.

Table 3 shows the fault coverage performance of the five models under different fault scales, reflecting the ability of each model to identify the fault propagation path under different node numbers. As the fault scale increases from 10 nodes to 100 nodes, the coverage of all models shows a downward trend, indicating that the larger the fault scale, the more difficult it is to accurately cover the fault propagation path. Among them, the proposed model performs the best, with an average coverage rate of 93.2%. When the fault scale is 10 and 50 nodes, the coverage rate is as high as 98.8% and 94.5%, respectively, which is significantly better than other models. The weakest performing GNN model has an average coverage rate of only 78.5%, and drops to 68.3% when the fault scale is 100 nodes. The T-GCN and GAT models follow, with average coverage rates of 89.4% and 86.7%, respectively, showing good robustness and stability. Overall, the proposed model maintains a high coverage effect at different fault scales.

TABLE 3. Fault coverage performance of five models under different fault scales

Fault Scale (Number of Nodes)	Proposed	T-GCN	GCN-LSTM	GAT	GNN
10	98.8%	96.5%	91.2%	94.7%	89.3%
30	97.2%	93.8%	86.7%	90.5%	83.6%
50	94.5%	90.1%	82.4%	87.3%	78.9%
80	89.7%	85.3%	77.6%	82.40%	72.5%
100	85.9%	81.2%	73.8%	78.6%	68.3%
Average Coverage Rate	93.2%	89.4%	82.3%	86.7%	78.5%

IV. CONCLUSION

To secure the high-quality communication and power stability essential for continuous industrial processes in energy-intensive sectors, this paper proposes a propagation path modeling method combined with GNN specifically designed to address the complex dynamic characteristics of fault propagation in power communication networks. By constructing a dynamic graph sequence based on node status and communication links, integrating position encoding and sliding time window technology, the T-GCN model is used to effectively capture the spatiotemporal dependencies in the fault propagation process. The model can output the fault probability of each node at future moments, generate dynamic propagation paths, and use the attention mechanism to identify key propagation nodes, thereby accurately locating the core hub of fault propagation. During the training process, the real propagation trajectory is used as a supervisory signal to significantly improve the accuracy and timeliness of propagation path prediction. Experimental results show that the proposed model performs stably and efficiently in node fault prediction tasks, with a precision of 0.92 and an average time error of 1.2 minutes, reflecting good time sensitivity. In terms of fault path identification, the coverage rate of key nodes is 90.5%; the overall average coverage rate reaches 93.2%; the propagation delay is effectively reduced by 31.2%. Even in the face of diverse fault types, the model's minimum F1-score remains above 0.84, reflecting its balance and robustness in accuracy and generalization. The framework proposed in this paper provides an effective tool for fault warning in power communication networks, which is crucial for securing the high operational reliability needed by energy-intensive sectors, and also offers a valuable reference for modeling and prediction of other spatiotemporal dynamic propagation problems.

AUTHOR CONTRIBUTIONS

Conceptualization – Lina Hu, Yi Tang and Lu Ye; methodology – Lina Hu, Yi Tang and Lu Ye; investigation – Lina Hu, Yi Tang and Lu Ye; writing-original draft preparation – Lina Hu and Yi Tang. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] H. Zhao, B. Yang, J. Cui, Q. Xing, J. Shen, F. Zhu, et al., "Effective fault scenario identification for communication networks via knowledge-enhanced graph neural networks," IEEE Transactions on Mobile Computing, vol. 23, no. 4, pp. 3243-3258, 2023, doi: 10.1109/TMC.2023.3271715.

[2] Z. Wan, L. Lin, Y. Huang, and X. Wang, "A graph neural network based fault diagnosis strategy for power communication networks," Journal of the Chinese Institute of Engineers, vol. 47, no. 3, pp. 273-282, 2024, doi: 10.1080/02533839.2024.2308236.

[3] A. Dwivedi and A. Tajer, "GRNN-based real-time fault chain prediction," IEEE Transactions on Power Systems, vol. 39, no. 1, pp. 934-946, 2023, doi: 10.1109/TPWRS.2023.3258740.

[4] H. Zhang, X. Ma, X. Liu, L. Li, and K. Sun, "Gnn-based power allocation and user association in digital twin network for the terahertz band," IEEE Journal on Selected Areas in Communications, vol. 41, no. 10, pp. 3111-3121, 2023, doi: 10.1109/JSAC.2023.3313192.

[5] Y. Gu, C. She, Z. Quan, C. Qiu, and X. Xu, "Graph neural networks for distributed power allocation in wireless networks: Aggregation over-the-air," IEEE Transactions on Wireless Communications, vol. 22, no. 11, pp. 7551-7564, 2023, doi: 10.1109/TWC.2023.3253126.

[6] Y. Gu, C. She, S. Bi, Z. Quan, and B. Vucetic, "Graph neural network for distributed beamforming and power control in massive URLLC networks," IEEE Transactions on Wireless Communications, vol. 23, no. 8, pp. 9099-9112, 2024, doi: 10.1109/TWC.2024.3358903.

[7] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," IEEE Transactions on Wireless Communications, vol. 22, no. 5, pp. 3554-3569, 2022, doi: 10.1109/TWC.2022.3219840.

[8] M. Lee, G. Yu, and H. Dai, "Decentralized inference with graph neural networks in wireless communication systems," IEEE Transactions on Mobile Computing, vol. 22, no. 5, pp. 2582-2598, 2021, doi: 10.1109/TMC.2021.3125793.

[9] J. T. de Freitas and F. G. F. Coelho, "Fault localization method for power distribution systems based on gated graph neural networks," Electrical Engineering, vol. 103, no. 5, pp. 2259-2266, 2021, doi: 10.1007/s00202-021-01223-7.

[10] X. Li, M. Chen, Y. Liu, Z. Zhang, D. Liu, and S. Mao, "Graph neural networks for joint communication and sensing optimization in vehicular networks," IEEE Journal on Selected Areas in Communications, vol. 41, no. 12, pp. 3893-3907, 2023, doi: 10.1109/JSAC.2023.3322761.

[11] J. Suárez-Varela, P. Almasan, M. Ferriol-Galmés, Rusek.K, GeyerF, and X. Cheng, "Graph neural networks for communication networks: Context, use cases and opportunities," IEEE network, vol. 37, no. 3, pp. 146-153, 2022, doi: 10.1109/MNET.123.2100773.

[12] J. Hu, W. Hu, J. Chen, D. Cao, Z. Zhang, and Z. Liu, "Fault location and classification for distribution systems based on deep graph learning methods," Journal of Modern Power Systems and Clean Energy, vol. 11, no. 1, pp. 35-51, 2022, doi: 10.35833/MPCE.2022.000204.

[13] W. Liao, D. Yang, Y. Wang, and X. Ren, "Fault diagnosis of power transformers using graph convolutional network," CSEE Journal of Power and Energy Systems, vol. 7, no. 2, pp. 241-249, 2020, doi: 10.17775/CSEEJPES.2020.04120.

[14] M. Lee, G. Yu, H. Dai, and G. Y. Li, "Graph neural networks meet wireless communications: Motivation, applications, and future directions," IEEE Wireless Communications, vol. 29, no. 5, pp. 12-19, 2022, doi: 10.1109/MWC.001.2200023.

[15] Y. Sun, J. Zhang, Y. Zhang, L. Yu, Q. Wang, and G. Liu, "Environment information-based channel prediction method assisted by graph neural network," China Communications, vol. 19, no. 11, pp. 1-15, 2022, doi: 10.23919/JCC.2022.11.001.

[16] J. Y. Kim, J. Seon, and S. H. Yoon, "Classification Method based on Graph Neural Network Model for Diagnosing IoT Device Fault," The Journal of the Institute of Internet, Broadcasting and Communication, vol. 22, no. 3, pp. 9-14, 2022, doi: 10.7236/JIIBC.2022.22.3.9.

[17] Y. Yang, J. Mei, Z. Zhang, Y. Long, A. Liu, Z. Gao, et al., "Lightweight fault prediction method for edge networks," IEEE Internet of Things Journal, vol. 11, no. 6, pp. 11117-11136, 2023, doi: 10.1109/JIOT.2023.3333293.

[18] Z. Li, Y. Tu, and Z. Ma, "Root cause analysis of anomalies based on graph convolutional neural network," International Journal of Software Engineering and Knowledge Engineering, vol. 32, no. 08, pp. 1155-1177, 2022, doi: 10.1142/S0218194022500395.

[19] W. Lee, Y. Lee J, and H. Kim, "Mobile device-centric approach for identifying problem spot in network using deep learning," Journal of Communications and Networks, vol. 22, no. 3, pp. 259-268, 2020, doi: 10.1109/JCN.2020.000008.

[20] X. Zhu, L. Zhao, J. Cao, and J. Cai, "Fault diagnosis of 5G networks based on digital twin model," China Communications, vol. 20, no. 7, pp. 175-191, 2023, doi: 10.23919/JCC.2023.00.040.