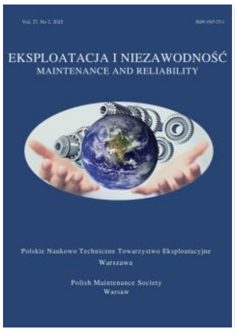


Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 27 (2025), Issue 2

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

Article citation info:
Gao Z, Liu W, Improved Graph Convolutional Neural Networks-based Cellular Network Fault Diagnosis, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2025: 27(2) http://doi.org/10.17531/ein/194672

Improved Graph Convolutional Neural Networks-based Cellular Network Fault Diagnosis

Indexed by: Web of Science Group

Zongzhen Gao[a,*], Wenlai Liu[a]

[a] School of Computer Science & Engineering, Linyi University, Linyi, 276000, China

Highlights

- This study uses extremal gradient enhancement to select the optimal feature subset.
- This study uses graph convolutional neural network to extract the fault depth feature.
- This technology utilizes Naive Bayes models for pre-diagnosis.

Abstract

To solve the problem of upstream and downlink interference in cellular networks, a graph convolutional neural networks-based novel fault diagnosis method for semi-supervised cellular networks is proposed. In the research design method, the extreme gradient enhancement technique is first used to enhance the fault diagnosis feature data of cellular networks. Then, the graph convolutional neural network is used to train and learn the fault diagnosis feature dataset of cellular networks, achieving fault diagnosis prediction of cellular networks. In the process of training the cellular network fault diagnosis model, data augmentation techniques were used to enhance the training level of the model, while Bayesian networks were used for pre diagnosis to improve the diagnostic accuracy of the modified model. The experimental results show that the cellular network fault diagnosis model constructed in the study can achieve a classification accuracy of 90% for training samples during training and testing, while other models can only achieve a maximum of about 85%. The model constructed by the research can achieve a diagnostic accuracy of over 90% in the practical application of cellular network fault diagnosis, while taking only 6 seconds. This algorithm can diagnose faults in complex cellular network environment, which has high accuracy and practicability, and can effectively improve user experience.

Keywords

Fault diagnosis, Naive Bayes, Knowledge data fusion, Graph Convolutional Neural Network

1. Introduction

The 21st century is known as the era of information technology. After more than 30 years of rapid development, mobile communication technology has been broadly utilized in various corners of society and has had a great influence on social development [1]. With the rapid growth of 5G networks and the imminent arrival of 6G networks, future mobile communication networks will become unprecedentedly heterogeneous and complex [2]. In such a network environment, ensuring network service quality, achieving rapid network fault diagnosis (NFD) with minimal cost and time has become an urgent issue that requires in-depth research. The obvious growing in the amount of device access in 5G networks has led to dense deployment of network nodes, increased overlap of base station coverage, and corresponding

(*) Corresponding author.
E-mail addresses: Z. Gao (ORCID: 0009-0003-6491-8179) gaozongzhen740909@163.com, W. Liu (ORCID: 0009-0006-1358-852X) liuwenlai@lyu.edu.cn,

increase in interference phenomena. The diversity of underlying devices makes the configuration of base station parameters more complex [3]. In such a complex network environment, how to achieve fast and accurate NFD has been an urgent issue that needs to be addressed. To address this challenge, more and more research is beginning to introduce machine learning methods into the field of NFD. Even when the network environment changes, machine learning algorithms can be trained or updated through new datasets to adapt to network fault diagnosis (FD) in the new environment. At present, the commonly used Cellular Network Fault Diagnosis (CNFD) methods face problems such as insufficient fault correlation analysis, low security, and low prediction accuracy when dealing with high complexity and large data networks. R. Levie et al. raised a CNFD method, which has acquired critical progress in FD in low complexity networks. However, when facing highly complex networks, the accuracy of FD rapidly decreased [4]. To further improve the FD performance of cellular networks, Hong et al. proposed a learning method that combines supervised and unsupervised learning. This method reduced the sample size and the accuracy of FD [5]. D. Zhu et al. proposed a method that combines convolutional neural networks (CNNs) and image restoration techniques, effectively reducing the FD cost of existing CNFD methods. However, this method did not improve the FD accuracy [6]. In order to ensure the stability and security of cellular network operation, research proposes to improve the existing fault diagnosis methods for cellular networks. The study used Extreme Gradient Boosting (XGBoost) to reduce the dimensionality of the operational data of cellular networks, and then constructed a fault diagnosis model for cellular networks using Graph Convolutional Neural Networks (GCNs). At the same time, data augmentation techniques were used to supplement the training data of the cellular network fault diagnosis model to enhance its training effectiveness.

The innovation of the research is to combine the XGBoost algorithm with the GCN algorithm, extract the fault characteristics of the cellular network by using the XGBoost algorithm, and then identify and extract the network fault parts through the GCN algorithm. The theoretical contributions of the research include innovation in FD methods: proposing a graph convolutional neural network (GCN)-based semi-supervised CNFD method, providing a new theoretical perspective. Feature selection (FS) and optimization: The XGBoost algorithm is used for FS, optimizing the feature subset and improving the diagnostic efficiency and accuracy of the model. Knowledge data fusion: A knowledge data fusion technology has been proposed, which enhances the model's understanding of complex data patterns by combining the pre diagnostic results of naive Bayesian models with domain expert knowledge. The technical contributions of the study include improvements to the GCN model, which improves the traditional GCN model by adjusting its structure and parameters to make it more suitable for CNFD tasks. Data augmentation technology: By applying GANs technology, the problem of insufficient labeled samples and uneven distribution of categories in real datasets has been solved. Practical application verification: The efficacy of the improved method was proved on real network datasets, demonstrating its high accuracy and practicality in actual CNFD. Algorithm efficiency and scalability: The merits of the raised method in efficiency and accuracy, as well as good scalability and repeatability, have been demonstrated through experiments on imbalanced datasets.

The first part of the article introduces the relevant research on NFD and network data collection. The second part introduces the GCN-based NFD algorithm and its improvements. The third part exhibits experiments and outcomes of validating the effectiveness of the algorithm using simulated and real data sets. The fourth part displays a comprehensive summary, and then identifies limitations and prospects.

2. RELATED WORK

NFD aims to monitor various nodes deployed in the network, collect network data generated during operation, and analyze them to detect abnormal states of nodes. Nowadays, many studies have applied graph theory and machine learning to the field of NFD in computer networks. M. S. Riaz et al. raised a framework that integrates CNN and image restoration techniques, which can perform root cause analysis. It demonstrated robustness to sparse minimization reports, base station locations, and fault types. Experimental results showed that this approach outperformed other learning-based models used in the test [7]. M. Chen et al. introduced an active learning-based FD approach that achieves excellent diagnostic performance while minimizing the need for labeled training samples, thereby reducing costs significantly. The key concept was to strategically choose the most informative unlabeled data for labeling and training purposes. Experimental results showed that to achieve the same diagnostic accuracy, this scheme only used fewer labeled training instances compared to existing non-active ones [8]. Y. Wang et al. reduced the need for labeled data by combining supervised and unsupervised learning. Experimental results showed that this method reduced the expensive cost of labeled data compared to traditional methods and achieved an FD accuracy of 99.08% based on simulation results [9]. K.M. Chen et al. introduced a diagnostic algorithm based on machine learning that leverages condition indicators. The algorithm ingeniously combined the original softmax network with SVM. Simulation outcomes denoted that the raised algorithm realized prominent diagnostic efficacy, surpassing traditional score-based methods [10]. M. S. Riaz et al. introduced an innovative framework that integrates CNN and XGBoost to diagnose multi-faults in base station networks. Performance assessment under realistic and extreme conditions revealed that the framework achieved an accuracy of 93%, surpassing existing FD solutions. Furthermore, it exhibited enhanced robustness in handling sparse reports [11]. To make sure the stable and safe operation of the lithium battery system, X. Hu et al. discussed the future development direction of the battery FD technology from the fault mechanism of the battery, the system sensor and the actuator. The findings denoted that the deep learning method was going to be the main method for FD of lithium battery systems [12].

Under the fast growth of the drive test, the collection process of real network data has become simple and convenient. However, the lack of a sufficient labeled data set for training FD models remains a challenging problem. A. Rizwan et al. proposed a data augmentation scheme [13]. R. W. L. Coutinho and A. Boukerche discussed transfer learning overcoming the need for massive high-quality data to train machine learning models in the Industrial Internet of Things (IoT). They also described the working principles and significant challenges faced by transfer learning systems in the industrial IoT. They divided the transfer learning systems in the industrial IoT into mechanical and network hierarchies and provided in-depth discussions on the design components and challenges of transfer learning systems in each proposed category [14]. L. Cerdà-Alabern et al. attempted to use anomaly detection technologies to detect hardware failure events. They applied four unsupervised machine learning methods based on different principles. Numerical results showed that all tested methods improved the performance in detecting gateway failures when non-traffic features were considered simultaneously [15].
S. Han et al. proposed a weighted generalized learning system where the weight assigned to each class depends on its sample quantity. To further improve the classification performance, they introduced an improved differential evolution algorithm to automatically optimize the generalized learning system and newly generated weights and parameters. Experiment outcomes said that this method achieved greater classifying accuracy than other methods on 20 imbalanced classification problems [16].
Y. Wang et al. proposed an ensemble learning-based self-organizing heterogeneous NFD system and used a synthetic minority oversampling technique to handle data imbalance. Considering the cost-sensitivity, they used rescaling methods to help the classifier differentiate the importance of different samples to minimize the overall loss. Additionally, to address the sparse data in small areas and dense deployment in heterogeneous networks, the solution provided a distributed diagnosis system that reduces communication costs [17]. T. Ahmad et al. found that the GCNs algorithm could realize the action recognition of the human body skeleton through multi-node or multi-lateral data. Therefore, the authors proposed an action recognition classification method based on GCN technology, and elaborated on the future development direction of GCN technology [18]. A. Z. Yonis et al. proposed a 5G communication network communication channel based on non orthogonal multiple access technology to improve the processing capability of wireless data stream large-scale data services in 5G communication systems. The findings denoted that the use of this technology can raise the anti-interference

ability of communication channels [19].

In summary, the academic community has discussed and researched network fault detection and diagnosis. Researchers have endeavored to incorporate cutting-edge technologies into the process of FD, yielding a series of solutions. Nevertheless, the growing uncertainty and intricacy of mobile communication networks render these solutions inapplicable to the present network environment. To better ensure the healthy operation of networks, a reliable and efficient NFD solution remains a research focus for future researchers. Thus, this study focuses on the precise diagnosis of 4G/5G network faults with limited labeled samples and proposes an improved CNFD supported by GCN. The application of this method in cellular networks is crucial as it not only demonstrates its necessity but also demonstrates the uniqueness and superiority of FD achieved through deep learning frameworks in the rapidly developing and diverse modern communication environment of network faults.

3. CNFD-BASED IMPROVED GCN

The study first proposes a CNFD based on GCN, which utilizes GCN for extracting graph data features and classifying nodes. To solve the problem of limited labeled samples and imbalanced class distribution in real data sets, the study further improves the GCN-based diagnostic method using knowledge and data fusion techniques. The improved method augments the real data set using Generative Adversarial Networks (GANs) to increase the sample size. Then, combining domain expert knowledge, the dataset is pre-diagnosed using the Naive Bayes method, and the generated association graph is used as prior knowledge to guide the GCN training. At last, the original GCN has been enhanced to allow for the calibration of the influence of prior diagnostic knowledge and training dataset scale on the model accuracy during the training.

3.1. GCN-BASED FAULT DIAGNOSIS

The cellular network structure includes massive base stations, user devices, and other elements, with complex interactions and dependencies between them. GCN can effectively simulate the dynamic relationships between these devices and model the propagation and impact of faults. The data in cellular networks is usually not arranged in a regular manner, but forms a complex graphical structure. The non-Euclidean data processing capability of GCN makes it perform better in such applications. The node information transmission scheme of CGN network is mainly based on content addressing mechanism, which locates and retrieves data through content identifiers (CID), rather than relying on traditional IP addresses. This scheme utilizes a caching mechanism, allowing nodes to store content replicas locally, thereby reducing duplicate transmissions, improving efficiency, and supporting offline access. Meanwhile, CGN networks support multipath transmission, enhancing the reliability and efficiency of data transmission. In addition, the distributed storage characteristics of CGN networks ensure high availability and redundancy of content. The discovery and sharing of node information are achieved through broadcasting or multicast mechanisms, while encryption and authentication mechanisms ensure the security of information transmission. The adaptability of CGN networks allows for dynamic adjustment of information transmission strategies based on network conditions and user needs, optimizing overall performance. Therefore, the study proposes to design the FD model of cellular network. The study proposes a CNFD based on GCN. Firstly, the XGBoost algorithm is applied to conduct FS on the original network parameter dataset. Then, the reduced dataset is transformed to obtain graph data in the format required by GCN. Finally, GCN is used to diagnose network parameter data for FD. Although using XGBoost can improve the performance of the model, this method may introduce bias or over-fitting. To address this potential issue, cross validation and regularization techniques are used to evaluate the generalization ability of selected features and adjust the parameters of XGBoost to alleviate over-fitting. The workflow of the raised algorithm is denoted in Figure 1. The process shown in Figure 1 first includes feature selection and data standardization processing. In the processing stage, the XGBoost algorithm is used to perform feature selection on the original network parameter dataset, in order to reduce the feature dimension and optimize the diagnostic efficiency and accuracy of the model. After feature selection, the dataset is converted into the graph data format required by GCN. Subsequently, a feature matrix and a label matrix are constructed, and adjacency matrices are generated by calculating similarity, which are collectively used to represent the topological relationships between network nodes. Next, the GCN model processes graph data through graph convolutional layers to aggregate and classify node features. In the training process of GCN, the cross entropy loss function is used to calculate the error, and the gradient descent method is used to optimize the model weights.

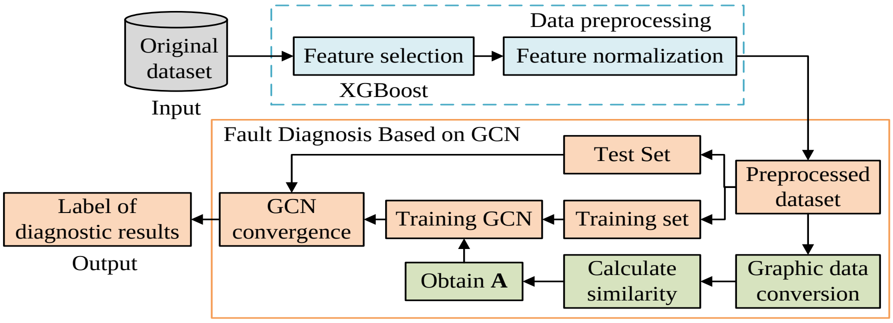


Figure 1. Flow chart of CNFD based on GCN.

In addition, in order to improve the generalization ability of the model, cross validation and regularization techniques were applied to evaluate the generalization performance of selected features, and the parameters of XGBoost were adjusted to alleviate overfitting problems. XGBoost is adopted to build a decision tree of multiple [20-21] and evaluate feature parameters' importance, which are ranked by the importance in a descending pattern. The optimal number is determined with the best combination conditions. The XGBoost's function of objective is indicated in Equation (1).

$$O^{(m)} = \sum_{i=1}^{N}\left(y_i, \hat{y}_i^{(m)}\right)^2 + \sum_{k=1}^{m}\left(\gamma T + \frac{1}{2}\lambda \sum_{j=1}^{C} \omega_j^2\right)\square \tag{1}$$

In Equation (1), $\left(y_i, \hat{y}_i^{(m)}\right)^2$ represents the loss function, which reflects the difference between the predicted network state labels $\hat{y}_i^{(m)}$ and the true ones y_i. i represents the index number of the data label. N is the amount of samples. $\sum_{k=1}^{m}\left(\gamma C + \frac{1}{2}\lambda \sum_{j=1}^{C} \omega_j^2\right)$ is the regularization term that limits the number of leaf nodes. λmeans regularized parameter, and γ means learn rate. C means the amount of leaf nodes, and ω_j is the weight on the j -th leaf node. Assuming the model structure is determined, the weights are able to be obtained by taking the means of using the objective function derivative which is planned as 0. By transforming the traversal over samples in the objective function to a traversal over leaf nodes and substituting the weights, the final objective function O^* can be obtained, as shown in Equation (2).

$$O^* = \gamma T - \frac{1}{2}\sum_{j=1}^{C}\frac{D_j^2}{H_j+\lambda} \tag{2}$$

In Equation (2), a smaller value indicates a closer prediction to the true result. There are $D_j = \sum_{i \in I_j} g_i$, and $H_j = \sum_{i \in I_j} h_i . g_i$ and h_i represent the first and second derivatives of the loss function for the i -th sample in the (m-1)-th round of models. Assuming d as the original parameters, and after FS, the feature parameters number is meant to be $d_0 (0 < d_0 < d)$. To prevent higher values from exerting undue influence over the entire training process of the GCN model, it needs to normalize them in the interval of [0, 1] using min-max normalization before using them. The proposed method requires mapping the reduced network parameter data set $\left\{ \begin{matrix} (x_1, y_1), (x_2, y_2), \cdots \\ , (x_l, y_l), (x_{l+1}, 0) \cdots, (x_n, 0) \end{matrix} \right\}$ to an un-directed graph. l is the number of labeled samples. In the data set, $x_i = \left[KPI_{i,1}, KPI_{i,2}, \cdots, KPI_{i,d_0}\right] \in \mathbb{R}^{d_0}$ represents the feature parameter vector under the condition that the current network state using d_0 KPIs. L represents the set of fault category labels for cellular networks, y_i belongs to the set$L$as class labels, and the label set $L = \{1,2,\cdots,c\}$ includes C network fault class labels. Assuming the first l data points are labeled withy_i, and the left data points are unlabeled data with class label y_i equal to 0. First, the feature matrix $X \in \mathbb{R}^{n \times d_0}$ is constructed from the data set, as shown in Equation (3).

$$\mathbf{X} = \begin{pmatrix} KPI_{1,1} & KPI_{1,2} & \cdots & KPI_{1,d_0} \\ \vdots & \vdots & \ddots & \vdots \\ KPI_{l,1} & KPI_{l,2} & \cdots & KPI_{l,d_0} \\ KPI_{l+1,1} & KPI_{l+1,2} & \cdots & KPI_{l+1,d_0} \\ \vdots & \vdots & \ddots & \vdots \\ KPI_{n,1} & KPI_{n,2} & \cdots & KPI_{n,d_0} \end{pmatrix} \tag{3}$$

To facilitate the calculation of cross-entropy loss in a manner that is conducive to the backward weight update of

GCN, it is necessary to have a label matrix in place to indicate the various label categories. The encoded class labels can be obtained in Table 1.

Table 1. Electronic hot encoding of some network fault types in the data set

Class label	Fault type	Code
1	uplink interference	010000
2	Downlink interference	001000
3	Air port failure	000100
4	Base station failure	000010
Class label	Fault type	Code
5	coverage hole	000001
6	Normal situation	100000

The unlabeled data is assigned zero vectors as class labels, and these label vectors are then combined to form a label matrix $Y \in \mathbb{R}^{n\times c}$ encompassing all the data. Finally, an adjacency matrix $F \in \mathbb{R}^{n\times n}$ is required for the representation of the edge relationships between nodes. This is done by calculating the similarity measure $s_{i,j}$, as shown in Equation (4).

$$s_{i,j} = \begin{cases} e\left(-\frac{\|x_i-x_j\|_2^2}{2\delta^2}\right), i \neq j \\ 0, otherwise \end{cases} \tag{4}$$

In Equation (4), δ is the Gaussian bandwidth parameter. When $s_{i,j}$ is greater than the manually set threshold α, the corresponding element in the adjacency matrix $A_{i,j}$ =1, otherwise it is set to 0. The specific value of the threshold α will be determined through experiments. The GCN-based NFD model treats CNFD as classifying tasks and uses GCN to classify the data samples. The GCN model's structure is denoted in Figure 2.

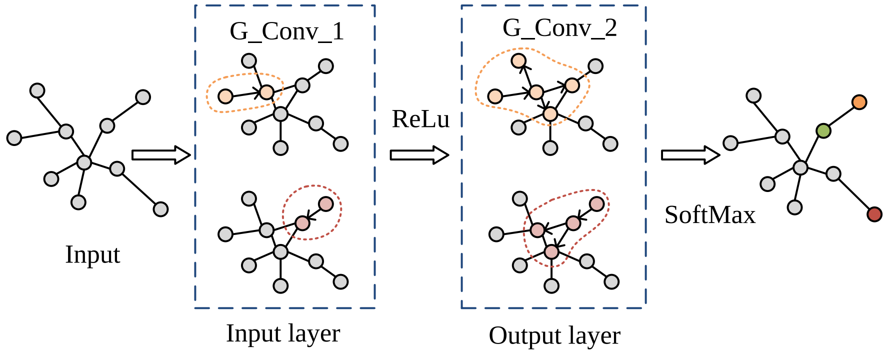


Figure 2. The graph convolutional neural networks model structure.

After obtaining the reduced network fault data set, the feature parameter vectors of the data samples in the dataset are first transformed as $n \times d_0$-dimension matrix $\boldsymbol{X}$. After this, an $n \times n$ -dimension adjacency matrix F is built. $\boldsymbol{X}$ and F are used to be GCN input. The forward propagation in GCN is indicated in Equation (5).

$$A^{(l+1)} = \sigma\left(\tilde{E}^{-\frac{1}{2}}\tilde{F}\tilde{E}^{-\frac{1}{2}}A^{(l)}J^{(l)}\right) \tag{5}$$

In Equation (5), σ means the representation of the activation function. I_N means the identity matrix. The purpose of this operation is for feature attribute information aggregation and its neighboring nodes' aggregation through graph convolution. $\tilde{E}$ here is identified as $\tilde{A}$'s degree matrix. $\tilde{F} = F + \boldsymbol{I}_N$. $A^{(l)}$ and $J^{(l)}$ are the input node feature matrix and the matrix for trainable weight, respectively. The output of GCN is $Z \in \mathbb{R}^{n\times c}$, which is recognized as a node feature matrix. Given that the research treats NFD as a node classification task, it is necessary for the final result to output a class label for every node. Accordingly, the network structure does not require the inclusion of fully connected layers, which are typical of traditional CNNs. It is sufficient to set the activation function on the final layer of graph convolutional layers (GCLs) to Softmax. The output node feature matrix is shown in Equation (6).

$$Z = SoftMax\left(\hat{F}A^{(1)}J^{(1)}\right) \tag{6}$$

In Equation (6), $J^{(1)}$ means the representation of the second layer weight matrix of graph convolution. The representation of the output result matrix Z is similar to the label matrix Y. The predicted label for the sample node a_i is $\tilde{y}_i = \underset{1\le j\le c}{argmax} Z_{i,j}$. In GCN training, the cross-entropy loss function $Loss$ needs to be calculated based on the labeled samples, whose error is back-propagated to improve the weight values in every GCL, employing gradient descent as the optimization technique, as shown in Equation (7).

$$Loss = -\sum_{i=1}^{l}\sum_{j=1}^{c} Y_{i,j}\, ln\, Z_{i,j} \tag{7}$$

3.2. Improvement Of Fault Diagnosis Method By Knowledge Data Fusion

After improving GCN using XGBoost method and constructing a new model of CNFD, it was found that the model had poor training results due to the lack of real fault data during the training process. Therefore, in order to improve the applicability of the improved GCN model in practical CNFD applications, it is proposed to use knowledge data fusion technology to enhance the training dataset of the improved GCN model, in order to strengthen the training level of the improved GCN. During the training process of GAN, simulation data will be generated based on the model results to improve the authenticity of the model output results. Therefore, the study proposes using GAN technology to expand the real dataset of the improved GAN model to enhance the training level of the improved GCN. The framework structure of using GAN to expand the training set of improved GCN with real samples is shown in Figure 3[22-23].

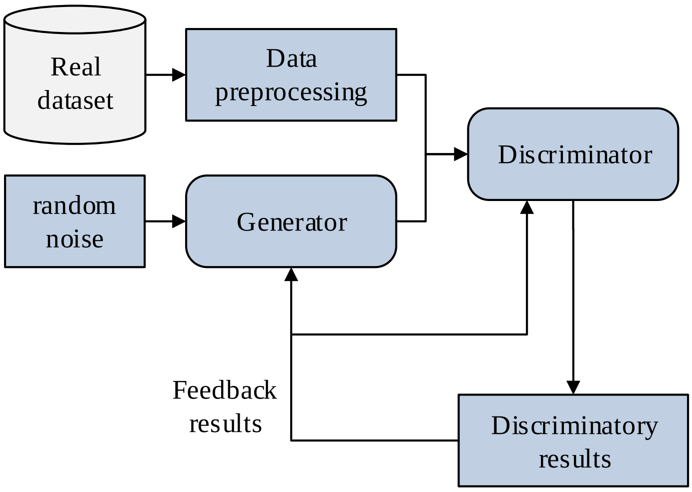


Figure 3. Dataset augmentation framework for the GAN model.

From Figure 3, GAN is a process that involves the interplay of two models, each competing with the other. During the training of GAN, there is a possibility that the generator may encounter difficulties due to gradient vanishing, particularly when the discriminator has already reached an optimal state. Without gradient information updates, the generator cannot further train, ultimately causing difficulties in the convergence of GAN [24-25]. To avoid weight restrictions on the discriminator network, the research uses the Wasserstein GAN-Gradient Penalty (WGAN-GP) algorithm, which introduces the Wasserstein distance and gradient penalty term (PT), to augment the data set. The loss function is displayed in Equation (8).

$$Loss_D = E_{\tilde{x}-P_g}[D(\tilde{x})] - E_{x-P_d}[D(x)] + \rho E_{\hat{x}-P_x}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{8}$$

In Equation (8), P_g means the distribution of the generated data by the generator, and $\rho \in 0, +\infty)$ means the penalty coefficient. $P_{\tilde{x}}$ represents the sampled data distribution in the PT, and the sample $\hat{x}$ is got by linear interpolation sampling between x and $\tilde{x}$, avoiding sampling the entire sample space. $\rho E_{\hat{x}-P_x}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$ means the PT, which forces the discriminator $\|\nabla_{\hat{x}} D(\hat{x})\|_2$to approach a gradient of 1 on the sampling point $\hat{x}$ as much as possible during the training of WGAN-GP, making the discriminator network satisfy the 1-Lipschitz constraint condition [26-27]. Before constructing the topological association graph, it is necessary to obtain the topological relationships between the data and calculate their similarities. However, the topological relationship graph constructed using clustering methods is currently rough and has little practical value in evolving the classification in accuracy of the GCN. In addition, this research also investigates the CNFD-based on the fusion of knowledge and data using actual network parameters to extend its application in dynamic network environments. The workflow of the improvement method is denoted in Figure 4.

In Figure 4, the improved algorithm includes two stages. In the first stage, a combination of Naive Bayes methods and expert knowledge is utilized to pre-diagnose network fault samples. Although the use of Naive Bayes models is somewhat innovative, it may oversimplify the complexity of handling cellular network data. Therefore, to overcome the limitations of Naive Bayes models in handling complex data, expert knowledge is introduced to more accurately capture and understand the patterns and features in cellular network data. In accordance with the outcomes of the preliminary diagnostic assessment, correlation graphs are employed to facilitate a supplementary pre-diagnosis of the GCN model. In stage two, the topological association graph generated in the preceding stage and the aforementioned training dataset are inputted into the GCN for the purposes of model training.

The original GCN has been enhanced in order to address the potential influence of pre-diagnosis prior knowledge and the scale of the training dataset on the accuracy of the GCN model during the training phase. The improved algorithm still uses the XGBoost for selecting the optimal feature subset of the data set. Assuming the pre-processed network fault data set is represented as $\{(x_1, y_1), (x_2, y_2), , (x_n, y_n)\}$, where $x_i = [KPI_{i,1}, KPI_{i,2}, \cdots, KPI_{i,M}]$ represents the feature parameter vector reflecting the network status under the current environment using M KPIs. In the NFD scenario, taking network fault $Y = \{y_1, y_2, \cdots, y_L\}$ as the parent node of the Naive Bayes model, and after FS, the KPI feature parameter variables in the dataset are taken as the child nodes $X_1, X_2, \cdots, X_M$. The principle of Naive Bayes is shown in Equation (9) [28-29].

$$P(y_i|X) \propto P(y_i) \prod_{M}^{j=1} P(x_j|y_i) \tag{9}$$

In Equation (9), $P(y_i|X)$ means the posterior probability, and $P(y_i)$ is the prior probability. x_j represents the specific value of the j -th feature parameter in X .

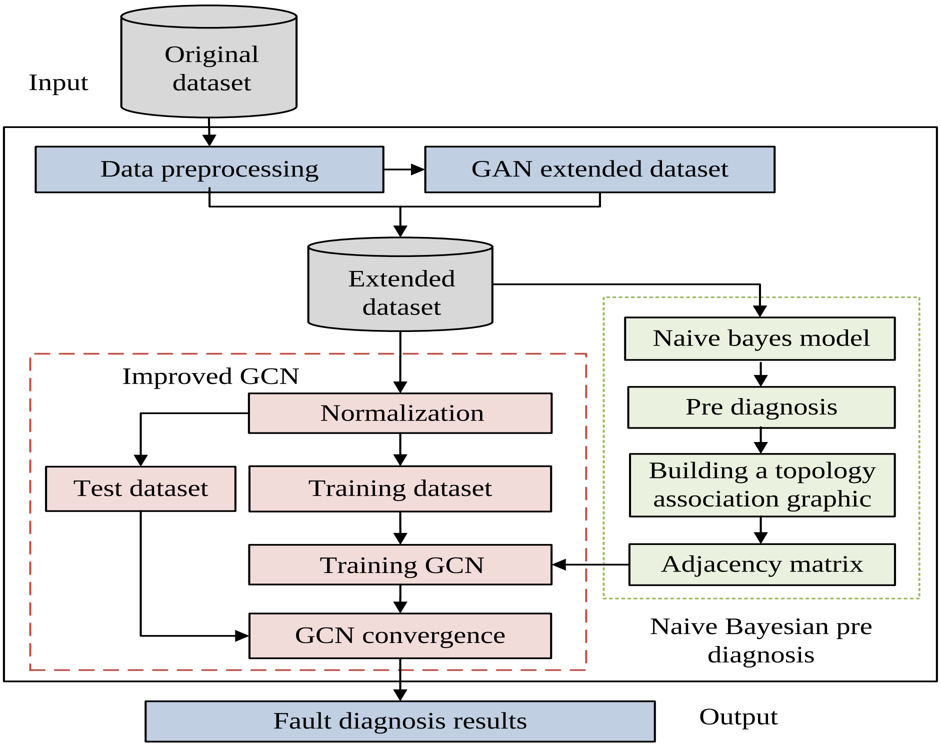


Figure 4. Flowchart of CNFD method based on knowledge and data fusion.

In accordance with NFD theory, the Naive Bayes model calculates the posterior probability distributions using the learned model for each defined set of fault causes and then selects the network fault type that maximizes the posterior probability as the current network fault $h^*(X)$. This is based on the input feature parameter vector that represents the network state. However, in practical situations, solving $h^*(X)$ involves multiplying multi-conditional probabilities. Thus, to prevent underflow errors, logarithmic form is used for calculation, as shown in Equation (10).

$$h^*(X) = argmax\left[\sum_{j=1}^{M} log\, P\left(x_j|y_i\right) + log\, P\left(y_i\right)\right] \tag{10}$$

In Equation (10), the terms $P(x_j|y_i)$ and $P(y_i)$ represent the evidence utilized in the Naive Bayes reasoning process. To prevent bias in probability calculation and the occurrence of zero probability values, the research uses Laplace smoothing to calculate the prior probability $P(y_i)$ and conditional probability $P\left(x_j|y_i\right)$, as shown in Equation (11).

$$\begin{cases} P(y_i) = \left(D_{y_i}+1\right)/(D_t+L) \\ P\left(x_j|y_i\right) = \left(D_{y_i,x_j}+1\right)/\left(D_{y_i}+S_j\right) \end{cases} \tag{11}$$

In Equation (11), D_t represents the total training samples, D_{y_i} represents the total amount of samples in the training set under the condition of fault y_i, and D_{y_i,x_j} represents the amount of samples with KPI parameter x_j in D_{y_i}. L represents the number of types of network failures, and S_j represents all possible values of KPI. Then, based on the discretized KPI attributes, the training data set is divided properly to train the Naive Bayes classifier [30]. The trained Naive Bayes model is then applied to classify the remaining data, obtaining the overall diagnosis result label set $\hat{C} = \{\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_N\}$ in the first stage, where $N$is the representation of data set total samples after augmentation using WGAN-GP. In the GCN-based FD method, spectral clustering is applied for the adjacency matrix of the dataset, but the topological correlation graph generated by this method is not completely accurate. Therefore, in the improved method, the study uses Naive Bayes combined with expert knowledge to obtain the pre-diagnostic outcome set $\hat{C}$ to build a topological correlation graph, namely the adjacency matrix $\boldsymbol{A}$. In set $\hat{C}$, data associated with a given network fault type will be linked to each other, whereas the data associated with distinct network fault types will not exhibit such interconnectivity, as denoted in Equation (12).

$$A_{i,j} = \begin{cases} 1, \hat{c}_i = \hat{c}_j \quad i \neq j \\ 0, otherwise \end{cases} \tag{12}$$

According to Equation (12), a data topology correlation graph is actually composed of a certain number of independent subgraphs. This particular graph structure can effectively incorporate the pre-diagnosis outcomes into the subsequent training of the GCN model and solve the problem of determining the graphic convolution layer number [31-32]. After defining the adjacency matrix, the GCN is improved based on the obtained topological association graph, and the improved GCN is used to complete the second stage of FD task, obtaining the final NFD result. $\widetilde{\boldsymbol{D}}$ means the representation of the degree matrix, where every element on the principal diagonal is the sum of all elements in the related row of $\widetilde{\boldsymbol{A}}$. Therefore, the formation of $\widetilde{\boldsymbol{D}}$ depends entirely on $\widetilde{\boldsymbol{A}}$. $\widetilde{\boldsymbol{A}}$ is determined directly by the adjacency matrix $\boldsymbol{A}$ and the identity matrix. Therefore, $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} = f(\boldsymbol{A}, \boldsymbol{I}_n)$ is utilized to improve the forward propagation process of the original GCN, as shown in Equation (13).

$$\mathbf{H}^{(l+1)} = \sigma\left(f(\mathbf{A}, \mathbf{I}_n)\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right) \tag{13}$$

In the original GCN model theory, the matrix $\boldsymbol{A}$ is found and identified via an identity matrix of the same size. In this research, a weight coefficient β is added on this basis, as shown in Equation (14).

$$\widetilde{\mathbf{A}} = \mathrm{A} + \beta\mathbf{I}_n \tag{14}$$

In Equation (14), β represents a weight coefficient that exhibits a positive correlation with train size, and its definition is shown in Equation (15).

$$\beta = 1 + re^r \tag{15}$$

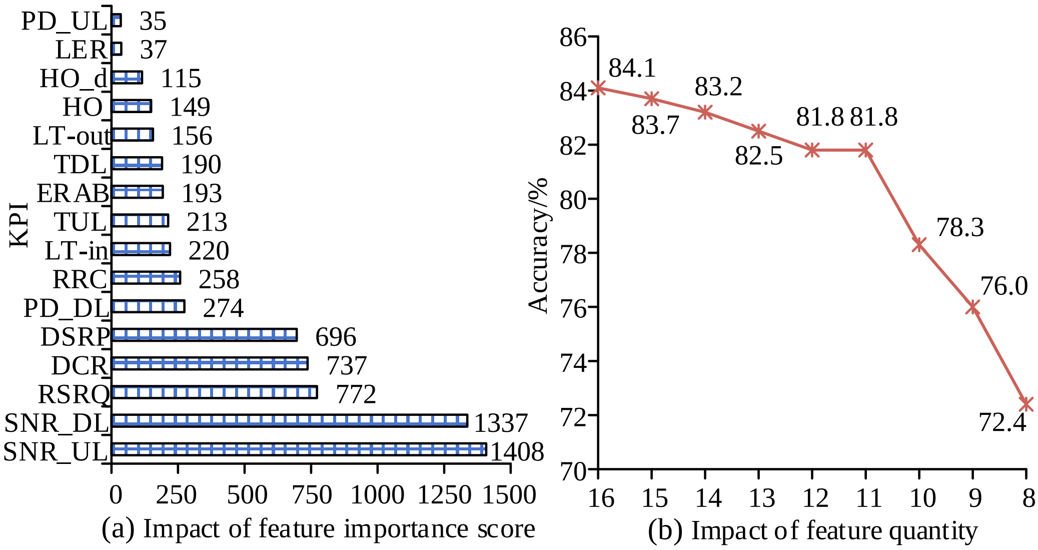
In Equation (15), the variable r denotes the labeled training to the total sample ratio. The enhanced GCN model continues to employ the cross-entropy loss function for loss calculation and updates its network parameters using error back-propagation. Thus, in the second stage, the final NFD task is completed using the improved GCN.

4. PERFORMANCE ANALYSIS OF IMPROVED GCN BASED FAULT DIAGNOSIS

To assess the effectiveness of the improved GCN-based FD, the research first built a network environment using simulation software and collected data generated in the network. Then, the effectiveness of the GCN-based FD algorithm was validated on the simulated data set. The research also used a real network data set collected in a certain city and augmented the data set using WGAN-GP to verify the effectiveness of the improved FD algorithm that combines knowledge data fusion in practical applications.

4.1. Graphic observations for thep erformance of gcn-based fault diagnosis

To assess the efficacy of the GCN-based FD method, a network environment was built using OPNET 18.6 network simulation software and collected data generated in the network. During a specific period of network operation, the simulation software set corresponding network faults and recorded the network parameter data generated during this period, while labeling them with the corresponding fault types. Finally, these network parameter data collected under different network fault scenarios were saved in CSV files for subsequent algorithm program reading. The simulated data set used in the research included six network state categories with a relatively high proportion of data samples. There were 3258 samples. First, the XGBoost algorithm was used to assign an importance score to each feature attribute, which were sorted from high to low according to their importance scores. The feature attributes in the data set included Handover Success Rate (HO), Handoff Delay (HO-d), Dropped Call Rate (DCR), Throughput_Up Link (TUL), Throughput_Down Link (TDL), and others. The highest score feature attributes obtained from sorting were selected as the dimension-reduced sample data attributes. This selection process ensured that the model accuracy remains relatively high. Figure 5 demonstrates the significance of feature attribute combinations and the predictive accuracy of the XGBoost model for varying attribute configurations.



(a) Impact of feature importance score (b) Impact of feature quantity

Figure 5. Impact of data features on fault diagnosis accuracy ((a) Impact of feature importance score, (b) Impact of feature quantity).

In Figure 5 (a), it can be seen that the Signal Noise Ratio-Down Link KPI has the highest level of importance, with a score of 1408. From Figure 5 (b), when all KPI features were retained, the model accuracy was the optimal. From Figure 5 (b), as the amount of feature attributes gradually decreased, the accuracy of the models lightly decreased. When the amount of features decreased to 11 or less, the accuracy of the model significantly decreased. In the sight of the trade-off between diagnostic accuracy and training complexity, the research chose the top 11 feature attributes from Figure 5(a) as the attributes for the dimension-reduced data set. Since the total number of defined network state categories in the data set was 6, the feature dimension here for the input of final was determined as 6. Consequently, the sizes of the convolutional filters were selected as 7 and 6, respectively. This meant that in the first GCL, $\boldsymbol{W}^{(0)} \in \mathbb{R}^{11\times 7}$ had a size of 7, and in the second GCL, $\boldsymbol{W}^{(1)} \in \mathbb{R}^{7\times 6}$ had a size of 6. For the over-fitting defect, a 0.25 probability dropout layer was added to the input data of each GCL. Therefore, the final structure parameters of the GCN are denoted in Table 2

Table 2. The structural parameters of the final GCN.

Amount of layers	1	2	3	4	5	6
Type	Inputting layer	Dropout layer 1	GCL 1	Dropout layer 2	GCL 1	Softmax layer
Output feature scale	3258×11	3258×11	3258×7	3258×7	3258×6	3258×6

Accuracy and Macro F1 were utilized as metrics to identify the efficacy of the FD model. To determine the value of the threshold and the depth of the GCN network, simulation experiments were conducted, and the results are shown in Figure 6.

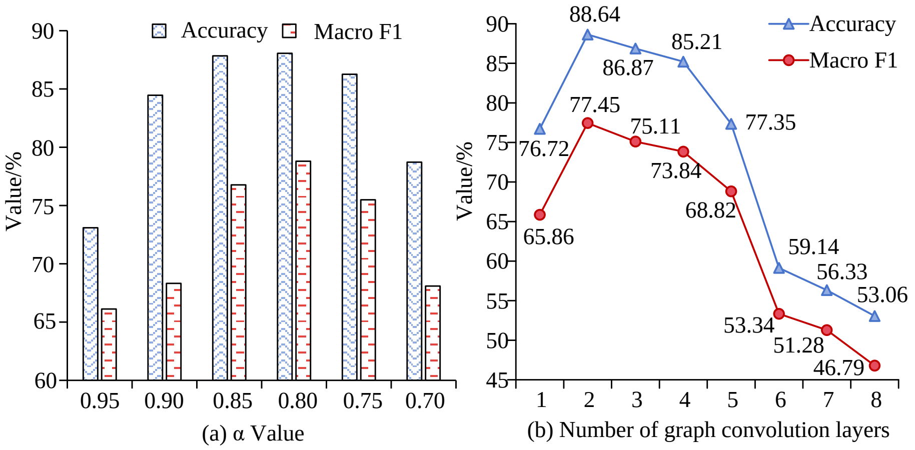


Figure 6. Parameter settings for GCN model ((a) α Value, (b) Number of graph convolution layers).

Figure 6 (a) shows the performance of GCN as a function of α. The change in value could be seen in the α. When the value was greater than 0.80, the accuracy of the model would vary with α. As the value increased, it decreased. At α 0.95, the accuracy of the model was only 74%. At α less than 0.80, the accuracy of the model would vary with α. At α 0.70, the model’s accuracy was only about 84%, compared to α 0.80, where the model accuracy was the highest, reaching around 88%, and the change in Macro F1 was consistent with the change in accuracy. Figure 6 (b) shows the variation of model performance with network depth. When the GCN depth was only two layers, the model performance was highest, with an accuracy of 88.64% and a Macro F1 of 77.45%. As the network depth continued to increase, the model performance began to decline. To assess the proposed one's effectiveness, this chapter compared the GCN-based CNFD model with the KNN model, decision tree model, RANK-SVM model, and improved BP neural network model through comparative experiments. Five sets of comparative experiments were performed.

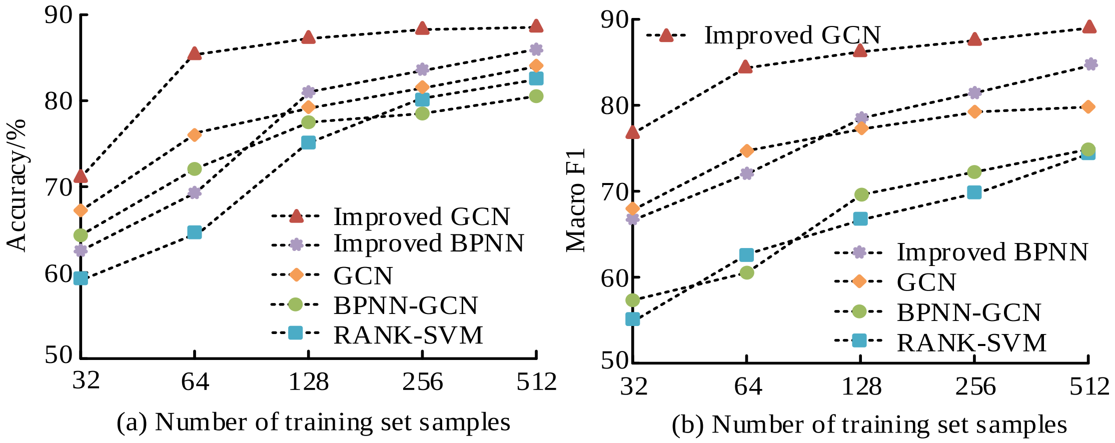
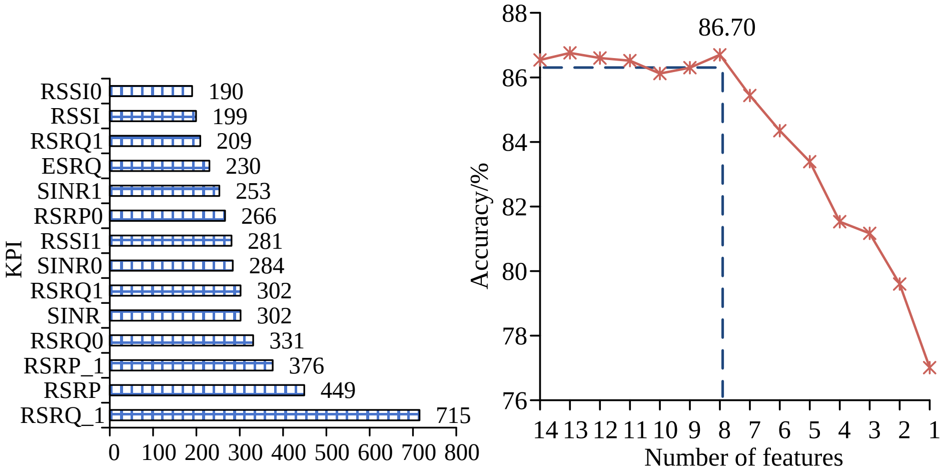


Figure 7. Comparison and analysis of training results of different algorithms ((a) accuracy comparison results, (b) Macro F1 comparison results).

Each of them included a subset of sample data covering various possible state types for networks, while the remaining data in the dataset served as the test set for observing the model's performance. Considering real-world scenarios, FD datasets typically consisted of major unlabeled samples and minor labeled ones. Therefore, the comparative experiments did not include training sets with more than 512 labeled samples. Figure 7 illustrates the diagnostic accuracy and Macro F1 scores of each algorithm under different training sets. Figure 7 (a) shows the variation of model accuracy with the amount of training set samples. Regardless of the amount of training set samples, the accuracy of the improved GCN algorithm was always higher than the other four algorithms’ performance. Even with only 64 samples in the training set, improved GCN still achieved an accuracy of over 80%. Figure 7 (b) shows the Macro F1 of the model. Similar to the results in Figure 7 (a), the improved GCN algorithm had a much higher Macro F1 than the other four algorithms. Intuitively, improved GCN had higher diagnostic performance than other algorithms under 5 different training sets. Simulation experiments showed that improved GCN could achieve good NFD accuracy even with a small amount of training samples.

4.2. Analysis of the practical application of GCN model in cellular network fault diagnosis

To avoid the problem of overly idealized diagnostic results caused by training algorithm models with simulated data sets, the performance analysis of the improved algorithm used a data set collected in a real network scenario through data collection on the roadside. This data set consisted of real user-side data collected by a enterprise in a certain city in August 2021, and it contained 817 labeled data samples. The results of feature importance ranking and FS experiments are shown in Figure 8.



(a) Importance score

(b) Accuracy of combining different quantitative features

Figure 8. Result of feature screening test ((a) feature importance score screening result, (b) feature quantity screening result).

Figure 8 (a) shows the importance scores of each KPI. RSSI had the lowest score, with only 190 points. Except for RSRQ_1, the importance scores of other KPIs were all below 500. The importance score of RSRQ_11 had reached over 700 points. The importance of different KPIs varied significantly. From Figure 8(b), under selecting 13 and 8 features, the proposed one's diagnostic accuracy was good to be accepted. However, to achieve the goal of FS, in the subsequent experiments, only the top 8 KPI parameters in the ranking in Figure 8(b) were selected as the feature-selected KPI parameters. To generate simulated data that conformed to the distribution of real data and solved the problems of imbalanced sample distribution and insufficient labeled data in some categories in the real network parameter data set, WGAN-GP was applied to augment the data set. When fitting the real network data in the "large inter-site distance" network fault scenario in the original dataset using WGAN-GP, the discriminator’s loss function is denoted in Figure 9.

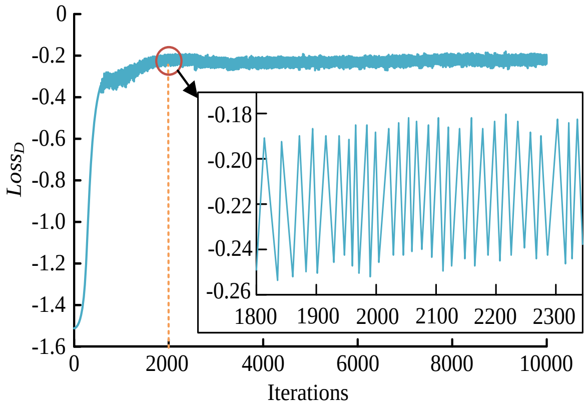


Figure 9. Loss function of the WGAN-GP fit to the real network data.

From Figure 9, the discriminator loss initially exhibited significant oscillations and then small oscillations after quickly converging, indicating that the model was in the learning phase and had not yet found the best solution direction. After approximately 2,000 iterations of training, the discriminator loss function reached a point of stability, indicating that the model in question had converged. The augmented data set was obtained through a process of integration, whereby generated simulation data was fused with the real data present in the original dataset, as demonstrated in Figure 10.

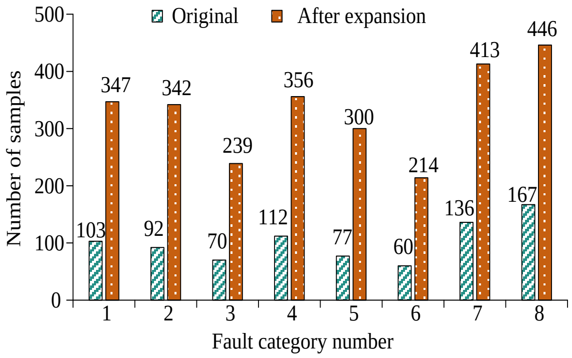


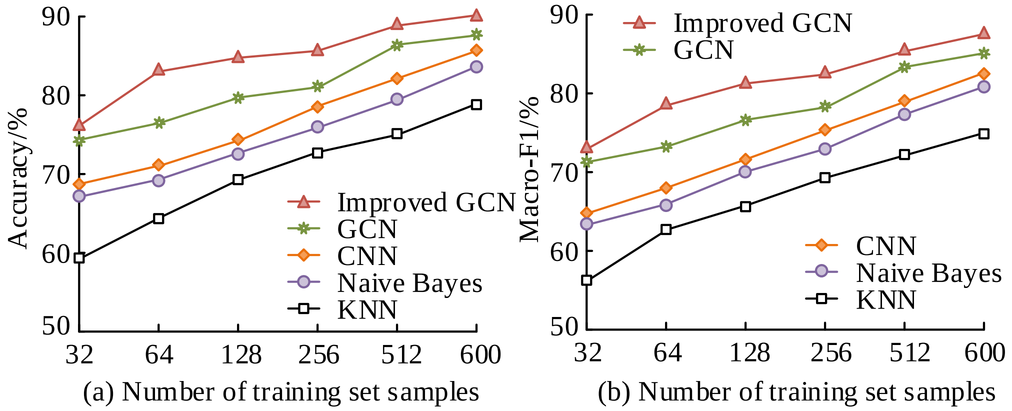
Figure 10. Data distribution of the expanded dataset using WGAN-GP.

In Figure 10, the original real data set was increased in size by approximately threefold through the application of WGAN-GP, resulting in a total of 2657 labeled data samples. Accordingly, the discretized data set may be readily used to calculate a likelihood function through statistical counting, thereby facilitating the first-stage pre-diagnostic classification. This is achieved by considering the frequency of KPI values under different network fault states in the training data set. The results of discretizing KPIs using expert knowledge are shown in Table 3.

Table 3. The result of discretization operation on feature attributes.

	1	2	3	4	5
RSRP/dBm	≤-115	(-115,-105]	(-155,-95]	(-95,-85]	>-85
RSRQ/dB	≤-20	(-20,-15]	(-15,-10]	>-10	/
RSSI/dBm	≤-100	(-100,-85]	(-85,-70]	(-70,-55]	>-55
SINR/dB	≤3	(3,10]	(10,15]	(15,25]	>25

GCNs are one of the most common variants of CNN, and it is widely used in many fields. K Guo et al. found that either the network depth was too deep or the number of iterations would lead to the model over-fitting phenomenon, and the value of learning rate and exit layer probability would directly affect the training result of the model [33-34]. According to the results of K Guo et al., in the comprehensive performance evaluation experiment of the FD model, the GCN model had a hidden layer depth of 2, the learning rate was 0.01, and the probability of exit layer was 0.25. The max training iteration times was set to 200, and the L2 regularization parameter was set to 110-5. Six experiments were performed with training set sizes of 32, 64, 128, 256, 512, and 600. Each experiment was repeated 10 times and the mean value was calculated to obtain the final result, as denoted in Figure 11.



(a) Number of training set samples

(b) Number of training set samples

Figure 11. Performance comparison and analysis of different algorithms ((a) accuracy comparison results, (b) Macro F1 comparison results).

From Figures 11(a) and 11(b), the raised algorithm achieved the highest accuracy and Macro F1 values compared to others in all experimental groups. The improved algorithm outperformed both the GCN-based and Naive Bayes-based NFD algorithms. Taking 600 iterations as an example, the proposed algorithm improved the accuracy and Micro F1 of the pure GCN by 2.2% and 1.8%. The improved algorithm combined the merits of these two algorithms and effectively addressed their demerits in a complementary manner. To in depth analyze the reference of the raised CNFD method in practical scenarios, the study used the fault types of weak signal, frequency band mismatch, operator service fault, and network roaming problems, and the findings are indicated in Table 4

Table 4. A practical application analysis based on the improved GCN NFD.

Type	Number of failures (time)	Detection frequency (time)	Number of warnings (time)	Detection success rate (%)
Weak signal	50	100	48	96.0
Frequency band mismatch	50	100	49	98.0
Operator malfunction	50	100	50	100.0
Network roaming failure	50	100	47	94.0

In Table 4, the research based on improved CNFD of GCN, the carrier, the fault detection rate could reach 100%, the diagnosis effect of network roaming problem was poor but also kept the fault detection rate above 90%. In the face of weak signal and frequency mismatch, the fault detection rate remained above 95%. The method could effectively realize the FD of the cellular network. A CNFD method based on IoT was proposed in reference [3], reference [4] presented a CNFD method based on CNNs, and reference [5] raised a CNFD method with the original GCN. To identify the repeatability and scalability of the raised algorithm, a non-equilibrium data set was constructed with the collected 2021 relevant data. In this dataset, the raised algorithm was compared with the three methods, and the outcomes are denoted in Figure 12.

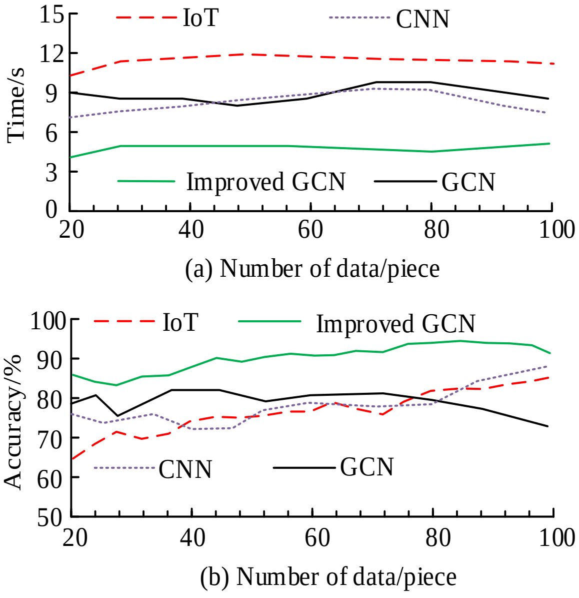


Figure 12. Comparison results of algorithm performance on imbalanced data sets ((a) comparison of fault diagnosis efficiency, (b) comparison of fault diagnosis accuracy).

Figure 12 (a) shows a comparison of the diagnostic efficiency of the four methods on the unbalanced dataset. In the unbalanced dataset, the FD efficiency of this algorithm was always higher than that of the other three methods, regardless of the amount of the data tested. The proposed algorithm's FD completion speed was the highest in 5 seconds, while the other three methods were the highest in 12 seconds. Figure 12 (b) indicates the outcomes of comparing the troubleshooting accuracy of the four methods on the unbalanced datasets. With the increase of test data, the accuracy of the FD method proposed by the study was continuously improved, which can reach about 92%, while the highest diagnostic accuracy of the other four methods was only 85%. On the unbalanced data set, this method was better than the other three FD methods in terms of accuracy and efficiency. The raised FD method had high efficiency and accuracy on both balanced and imbalanced data sets, with good scalability and repeatability. The proposed method combined GAN technology and expanded the training data set through GAN technology, so the proposed CNFD method performed well in both balanced and non-balanced data sets.

5. CONCLUSION

In the current cellular network environment, NFD has become particularly critical due to the high complexity of network node deployment and the frequent occurrence of faults. A CNFD based on GCN was invented in this study. To achieve efficient and accurate FD in densely populated and structurally complex cellular network environments, the study further improved the diagnostic method by combining knowledge and data fusion techniques. On the simulated data set, the GCN-based method achieved an average accuracy of 84.00%, which was significantly higher than the performance of other algorithms. In the real-world data set, the diagnostic method based on knowledge and data fusion improvement also achieved an accuracy of 84.33%. The experiment outcomes showed that the proposed one exhibited greater diagnostic accuracy and applicability compared to traditional methods, and could effectively cope with complex cellular network environments. The accuracy of the model has been improved after data expansion. In addition, the study also explored the impact of weight coefficients on diagnostic accuracy and found that as the amount of training samples grew, the effectiveness of the GCN model steadily improved. The CNFD method based on improved GCN performed well in terms of accuracy and reliability, especially suitable for scenarios with limited data labeling. The research results not only provide new ideas for CNFD but also provide efficient support tools for network maintenance and fault prevention, which is greatly significant for raising the healthy operation of mobile communication networks. Although the study used real data in the validation experiment of the improved method, which can reflect the real network situation to some extent, the network failures are limited to the coverage faults because the data are measured by the user terminal. Future research can explore the construction of more comprehensive network fault data sets, which will allow for a wider range of fault types to be covered and further improve FD techniques to adapt to constantly changing network conditions.

Reference

1. Porch J B, Heng Foh C, Farooq H, Imran A. "Machine learning approach for automatic fault detection and diagnosis in cellular networks," IEEE Int. Black Sea Conf. Commun. Netw., 2020; 2020(2020): 1-5, https://doi.org/10.1109/BlackSeaCom48709.2020.9234962.
2. Shafique K, Khawaja B A, Sabir F, Qazi S, Mustaqim M. "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," IEEE Access, 2020; 8(19): 23022-23040, https://doi.org/10.1109/ACCESS.2020.2970118.
3. Wu W, Peng M, Chen W, Yan S. "Unsupervised deep transfer learning for fault diagnosis in fog radio access networks," IEEE Internet Things J., 2020; 7(9): 8956-8966, https://doi.org/10.1109/JIOT.2020.2997187.
4. Levie R, Huang W, Bucci L, Bronstein M, Kutyniok G. "Transferability of spectral graph convolutional neural networks," J. Mach. Learn. Res., 2021; 22(1): 12462-12520, https://doi.org/10.5555/3546258.3546530.
5. Hong D, Gao L, Yao J, Zhang B, Plaza A, Chanussot J. "Graph convolutional networks for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., 2020; 59(7): 5966-5978, https://doi.org/10.1109/TGRS.2020.3015157.
6. Zhu D, Zhang F, Wang S, Wang Y, Cheng X, Huang Z, et al. "Understanding place characteristics in geographic contexts through graph convolutional neural networks," Ann. of Amer. Assn. Geogr., 2020; 110(2): 408-420, https://doi.org/10.1080/24694452.2019.1694403.
7. Riaz M S, Qureshi H N, Masood U, Rizwan A, Abu-Dayya A, Imran A. "Deep learning-based framework for multi-fault diagnosis in self-healing cellular networks," WCNC, 2002; 2022(2022): 746-751, https://doi.org/10.1109/WCNC51071.2022.9771947.
8. Chen M, Zhu K, Wang R, Niyato D. "Active learning-based fault diagnosis in self-organizing cellular networks," IEEE Commun. Lett., vol. 24, no. 8, pp. 1734-1737, Aug. 2020, https://doi.org/10.1109/LCOMM.2020.2991449.
9. Wang Y, Ruan Y, Tang Y. "Intelligent fault diagnosis method for mobile cellular networks," 2021 IEEE Globecom Workshops (GC Wkshps), 2021; 2021(2021): 1-6, https://doi.org/10.1109/GCWkshps52748.2021.9682015.
10. Chen K M, Chang T H, Wang K C, Lee T S. "Machine learning based automatic diagnosis in mobile communication networks," IEEE Trans. Veh. Technol., 2019; 68(10): 10081-10093, https://doi.org/10.1109/TVT.2019.2933916.
11. Riaz M S, Qureshi H N, Masood U, Rizwan A, Abu-Dayya A, Imran A. "A hybrid deep learning-based (HYDRA) framework for multifault diagnosis using sparse MDT reports," IEEE Access, 2022; 10(6): 67140-67151, https://doi.org/10.1109/ACCESS.2022.3185639.
12. Hu X, Zhang K, Liu K, Lin X, Dey S, Onori S. "Advanced fault diagnosis for lithium-ion battery systems: A review of fault mechanisms, fault features, and diagnosis procedures," IEEE Ind. Electron. Mag., 2020; 14(3): 65-91, https://doi.org/10.1109/MIE.2020.2964814.
13. Rizwan A, Abu-Dayya A, Filali F, Imran A. "Addressing data sparsity with GANs for multi-fault diagnosing in emerging cellular networks," ICAIIC, 2022; 2022(2022): 318-323, https://doi.org/10.1109/ICAIIC54071.2022.9722696.
14. Coutinho R W L, Boukerche A. "Transfer learning for disruptive 5G-Enabled industrial internet of things," IEEE Trans. Ind. Inform., 2022; 18(6): 4000-4007, https://doi.org/10.1109/TII.2021.3107781.
15. Cerdà-Alabern L, Iuhasz G, Gemmi G. "Anomaly detection for fault detection in wireless community networks using machine learning," Comput. Commun., 2023; 202(3): 191-203, https://doi.org/10.1016/j.comcom.2023.02.019.
16. Han S, Zhu K, Zhou M, Liu X. "Evolutionary weighted broad learning and Its application to fault diagnosis in self-organizing cellular

networks," IEEE Trans. Cybern., 2023; 53(5): 3035-3047, https://doi.org/10.1109/TCYB.2021.3126711.

17. Wang Y, Zhu K, Sun M, Deng Y. "An ensemble learning approach for fault diagnosis in self-organizing heterogeneous networks," IEEE Access, 2019; 7(6): 125662-125675, https://doi.org/10.1109/ACCESS.2019.2925566.
18. Ahmad T, Jin L, Zhang X, Lai S, Tang G, Lin L. "Graph convolutional neural network for human action recognition: A comprehensive survey," IEEE Trans. Artif. Intell., 2021; 2(2): 128-145, https://doi.org/10.1109/TAI.2021.3076974.
19. Yonis A Z, Nawaf A. "Investigation of evolving multiple access technologies for 5G wireless system," 2022 8th Int. Eng. Conf. Sustain. Technol. Dev. (IEC), Erbil, Iraq, 2022; 2022(2022): 118-122, https://doi.org/10.1109/IEC54822.2022.9807471.
20. Zhang Y, Zhang X, Sun Y. "Unsupervised fault diagnosis platform implementation for self-healing in cellular networks," 2020 IEEE Inform. Commun. Technol. Conf. (ICTC), 2020; 2020(2020): 192-197, https://doi.org/10.1109/ICTC49638.2020.9123269.
21. Li J, Zhu K, Zhang Y. "Knowledge-assisted few-shot fault diagnosis in cellular networks," 2022 IEEE Globecom Workshops (GC Wkshps), 2022; 2022(2022): 1292-1297, https://doi.org/10.1109/GCWkshps56602.2022.10008527.
22. Ren L, Jia Z, Wang T, Ma Y, Wang L. "LM-CNN: A cloud-edge collaborative method for adaptive fault diagnosis with label sampling space enlarging," IEEE Trans. Ind. Inform., 2022; 18(12): 9057-9067, https://doi.org/10.1109/TII.2022.3180389
23. Valdiviezo-Diaz P, Ortega F, Cobos E, Lara-Cabrera R. "A collaborative filtering approach based on naïve bayes classifier," IEEE Access, 2019; 7(8): 108581-108592, https://doi.org/10.1109/ACCESS.2019.2933048.
24. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip S Y. "A comprehensive survey on graph neural networks," IEEE Trans. Neural Netw. Learn. Syst., 2020; 32(1): 4-24, https://doi.org/10.1109/TNNLS.2020.2978386.
25. Bothe S, Masood U, Farooq H, Imran A. "Neuromorphic AI empowered root cause analysis of faults in emerging networks," 2020 IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom), 2020; 2020(2020): 1-6, https://doi.org/10.1109/BlackSeaCom48709.2020.9235002.
26. Hashmi G, Aljohani K, Kamarudin J. "Intelligent fault diagnosis for online condition monitoring of MV overhead distribution networks," 2022 4th IEEE Int. Conf. Appl. Autom. Ind. Diagn. (ICAAID), 2022; 1(3): 1-5, https://doi.org/10.1109/ICAAID51067.2022.9799512.
27. Ruan Y, Wang Y, Tang Y. "An intelligent cell outage detection method in cellular networks," 2021 16th IEEE Int. Conf. Comput. Sci. Educ. (ICCSE), 2021; 2021(2021): 548-553, https://doi.org/10.1109/ICCSE51940.2021.9569511.
28. Wang X, Lin H, Zhang H, Miao D, Miao Q, Liu W. "Intelligent drone-assisted fault diagnosis for B5G-enabled space-air-ground-space networks," IEEE Trans. Netw. Sci. Eng., 2020; 8(4): 2849-2860, https://doi.org/10.1109/TNSE.2020.3043624.
29. Riaz M S, H N Qureshi, U Masood, A Rizwan, A Abu-Dayya, A. Imran. "A hybrid deep learning-based (HYDRA) framework for multifault diagnosis using sparse MDT reports," IEEE Access, 2022; 10(6): 67140-67151, https://doi.org/10.1109/ACCESS.2022.3185639.
30. Liu Z, Zhang J, He X, Zhang Q, Sun G, Zhou D. "Fault diagnosis of rotating machinery with limited expert interaction: A multicriteria active learning approach based on broad learning system," IEEE Trans. Control Syst. Technol., 2022; 31(2): 953-960, https://doi.org/10.1109/TCST.2022.3200214.
31. Mismar F B, Hoydis J. "Unsupervised learning in next-generation networks: Real-time performance self-diagnosis," IEEE Commun. Lett., 2021; 25(10): 3330-3334, https://doi.org/10.1109/LCOMM.2021.3101058.
32. Zhang T, Zhu K, Niyato D. "Detection of sleeping cells in self-organizing cellular networks: An adversarial auto-encoder method," IEEE Trans. Cogn. Commun. Netw., 2021; 7(3): 739-751, https://doi.org/10.1109/TCCN.2021.3051326.
33. Cilínio M, Pereira M, Duarte D, Mata L, Vieira P. "Explainable fault analysis in mobile networks: A SHAP-based supervised clustering approach," 2023 16th Int. Conf. Signal Process. Commun. Syst. (ICSPCS), 2023; 2023(2023): 1-9, https://doi.org/10.1109/ICSPCS58109.2023.10261152.
34. Guo K, Hu Y, Qian Z, Liu H, Zhang K, Sun Y, et al. "Optimized graph convolution recurrent neural network for traffic prediction," IEEE Trans. Intell. Transp. Syst., 2021; 22(2): 1138-1149, https://doi.org/10.1109/TITS.2019.2963722.