Hindawi
Security and Communication Networks
Volume 2024, Article ID 9759083, 1 page
https://doi.org/10.1155/2024/9759083

*Retraction*

# Retracted: Path Selection Strategy of Communication Network Based on Graph Convolutional Neural Network

**Security and Communication Networks**

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

## References

[1] X. Zhang, "Path Selection Strategy of Communication Network Based on Graph Convolutional Neural Network," *Security and Communication Networks*, vol. 2022, Article ID 9548441, 14 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 9548441, 14 pages
https://doi.org/10.1155/2022/9548441

*Research Article*

# Path Selection Strategy of Communication Network Based on Graph Convolutional Neural Network

**Xiaojun Zhang**

*College of Software Technology, Henan Finance University, Zhengzhou 450000, China*

Correspondence should be addressed to Xiaojun Zhang; castorly@hafu.edu.cn

Received 14 April 2022; Accepted 24 June 2022; Published 18 July 2022

Academic Editor: Zhiping Cai

In order to improve the efficiency of communication network path selection, this paper combines the graph convolutional neural network to formulate the communication network path selection strategy and selects the enhanced decoding algorithm as the fixed-point decoding algorithm. For the quantization scheme based on the enhanced decoding algorithm, the selection of the fixed-point integer bit width of each operation variable in the SISO decoder is analyzed and determined by the method of statistical characteristic analysis. Moreover, this paper calculates the normalized threshold value according to the proposed state metric normalization scheme. In addition, this paper constructs an intelligent communication path selection system. Through research, it can be seen that the communication network path selection system based on graph convolutional neural network proposed in this paper can effectively improve the effect of multicommunication path selection.

## 1. Introduction

In recent years, with the rapid development of mobile communication technology, the cost of intelligent sensing devices such as smartphones and tablet computers has become lower and lower, and the communication capabilities have become stronger and stronger. The mobile ad hoc network (MANET) is a network composed of the above-mentioned smart devices, which has also developed rapidly and is widely studied in the field of network and its communication. The biggest feature of the MANET is that it can be self-discovered, configured, and self-organized, so it has the advantages of not requiring infrastructure support, highly dynamic, and mobile communication. Therefore, it has been widely used in marine organisms, forest fires, biomedicine, environmental monitoring, and other highly dynamic extreme environments that cannot be perceived in real time by humans and provides information collection, transmission, and processing services in extreme environments. For example, in-vehicle sensor network collects data through in-vehicle sensors or mobile phone devices, has mobility and can effectively cover the collection area, and is suitable for large-scale urban data collection. Wildlife tracking network collects animal behavior, physical condition, and movement information by deploying sink nodes, static nodes, and mobile nodes. In addition, the hybrid monitoring network in smart healthcare automates common medical processes to provide location, status, and tracking information for patients and assets. In the above-mentioned practical ad hoc network applications, the communication signal is attenuated due to the constant movement of nodes, the sparse network topology, and the occlusion of obstacles, which in turn makes the traditional MANET communication mode unable to operate normally. Ultimately, the network cannot be connected most of the time. Because MANET can only determine the next hop node after establishing the route between the communication endpoints, it needs to ensure that at least one link exists when the message is transmitted, and otherwise, the routing protocol of MANET cannot find the destination node.

When designing the algorithms and protocols of wireless ad hoc networks, the inherent characteristics of wireless ad hoc networks must be considered, and existing design ideas and methods for wired networks cannot be copied, so as to effectively support wireless ad hoc network applications. Algorithms and protocols in wireless ad hoc networks must have the characteristics of distributed autonomy, computational and storage efficiency, scale adaptability, and scalability. On the one hand, the wireless ad hoc network does not need to rely on any preset network facilities. It can be deployed at anytime and anywhere. There is usually no central control point similar to the base station in the network, and the nodes have equal status. The nodes coordinate their behaviors through layered protocols and distributed algorithms, quickly and automatically form an independent network, and support the dynamic joining and leaving of nodes. On the other hand, compared with personal computers, terminals in wireless ad hoc networks usually have small memory, low CPU processing capacity, and limited power supply, which requires that the software algorithms designed for wireless ad hoc networks must be simple and practical. Computational and storage costs are low. In addition, the scale of wireless ad hoc networks is usually large, ranging from dozens of nodes to thousands of nodes. The communication between any node in wireless ad hoc networks needs to involve multiple intermediate nodes. Especially when the network is large, it will inevitably have problems in communication security, load balancing, and other aspects, and put forward higher requirements for algorithms and protocols.

This paper combines the graph convolutional neural network to formulate the communication network path selection strategy, constructs an intelligent system, and improves the scientificity of the communication network path selection strategy.

## 2. Related Work

Graph data mining mainly includes graph clustering and subgraph pattern mining. The main researches on clustering algorithms in uncertain graphs are hierarchical clustering algorithm and K-nearest neighbor query algorithm [1]. A heuristic stochastic algorithm for solving the correlation clustering problem is proposed which can be used to solve the uncertain graph clustering problem [2]. An approximate algorithm for computing fuzzy clustering of uncertain graphs is proposed [3]. The research contents of subgraph pattern mining mainly include frequent subgraph patterns and dense subgraph patterns. Frequent subgraph pattern mining mainly studies the frequent subgraph pattern mining under the conditions of expectation and probability [4]. The goal of expected frequent subgraph pattern mining on uncertain graphs is to mine subgraph patterns with high expected support from the uncertain graph database, while the goal of probabilistic frequent subgraph pattern mining is to mine probabilistic frequent subgraph patterns in the uncertain graph database. *Subplot Mode.* Literature [5] proposes a search space pruning technique and an efficient expected support calculation method to improve efficiency. Literature [6] proposes the RAKING algorithm which combines random walk and uncertainty of uncertain mode. Dense subgraph mining mainly includes dense vertex subset mining and top-k maximal clique mining [7]. Graph clustering algorithms have many applications in communication networks and social networks.

In graph theory, the shortest path problem is a classical problem that has been widely studied and has a wide range of applications in transportation networks, communication networks, and social networks. In uncertain graphs, the research on the shortest path problem mainly includes the shortest path problem under probabilistic semantics and the shortest path problem under expectation semantics [8]. Literature [9] defines the shortest path problem under probabilistic semantics and proposes a shortest path algorithm based on probability threshold. Literature [10] defines the expected shortest path problem and proposes a random sampling approximation algorithm.

*Maximum Flow Problem.* The maximum flow problem is a classic combinatorial optimization problem in operations research, which is widely used in transportation networks, water supply networks, and financial systems. The Edmonds–Karp algorithm is a classic algorithm for solving the problem of determining the maximum flow in a graph. In uncertainty graphs, the most reliable maximum flow problem is mainly studied [11]. Literature [12] proposed a simple path-based algorithm and an algorithm based on space partitioning. Literature [13] proposed a state space partitioning algorithm based on probability and cutset double filtering. Literature [14] proposed three algorithms, which are NWCE algorithm based on negative weight community elimination, SPEA-t algorithm based on time constrained priority single ring elimination, and SPEA-p algorithm based on probability threshold constrained priority single ring elimination.

Minimum spanning tree has a wide range of applications. For example, it has applications in urban traffic construction and network design. Literature [15] studied the reliability of the minimum spanning tree in uncertain graphs and proposed a reliability calculation method based on the idea of edge set partitioning and optimized the method with the union search set. Literature [16] proposed an optimal spanning tree and the concept of suboptimal spanning tree; literature [17] studied the top-k spanning tree in uncertain graphs and proposed a PTopK query algorithm based on spanning tree filtering and a PTopK query algorithm based on combined filtering.

Literature [18] proposed an algorithm for dynamically modifying honeynet data. By dynamically sorting and modifying the data in the honeynet that may be accessed by the attacker to mislead the attacker, the deception ability of the honeynet is enhanced, so as to deceive the intruder who intends to destroy the completeness of the data. In addition, by deploying a honeynet transparent to attackers in the honeynet, the honeynet is separated from the business network where the honeynet is deployed, so that the traffic in the honeynet can be dynamically controlled according to the security requirements. Literature [19] proposes a software-defined honeynet (SDH) based on the idea of SDN. This honeynet can find the bottleneck of the link by calculating the relevant parameters of each link in the honeynet and use the SDN controller to dynamically Therefore, a dynamically generated fake honeynet topology can be presented to the attacker in the honeynet. Literature [20] proposes an SDN virtual honeynet for network attack situations. The honeynet

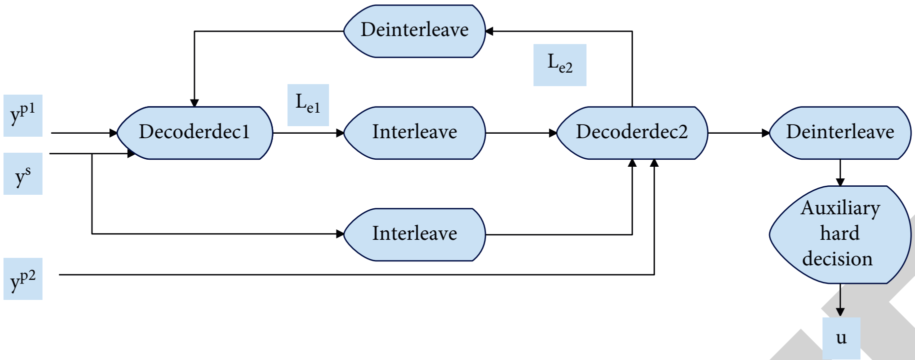

Figure 1: Schematic diagram of iterative decoding structure of turbo code.

system combines mimic defense and adopts SDN technology to realize flexible and dynamic control of honeynet traffic and realize the separation of honeynet data control layer and forwarding layer, which enhances the scalability and maintainability of the honeynet. At the same time, the validity of the dynamic transformation mechanism of the virtual honeynet is verified by using game theory.

## 3. Fixed-Point Turbo Code Decoding and Its Implementation

In the communication network constructed in this paper, in order to achieve better decoding performance of turbo codes, the decoding of component codes must use soft-input and soft-output decoding algorithms. Thus, the exchange of soft information between the component decoders in the iterative decoding process is realized. The structure of turbo code iterative decoder is shown in Figure 1.

The basic decoding process is that the received code sequence is demultiplexed, and the demodulated systematic bits $y^s$, check bits $y^{pl}$, and the deinterleaved outer information given by the second component decoder dec2 in the previous iteration are sent to the first component decoder dec1. The extrinsic information generated after decoding by dec1 is interleaved by the interleaver as a priori information of the second component decoder dec2 and sent to dec2 together with the interleaved demodulated systematic bits and the check bit $y^{p^2}$. The outer information generated by dec2 is deinterleaved and sent to the first component decoder decl for the next round of iterative decoding. When the number of iterations is completed or the stop iteration decision condition is reached, the log-likelihood ratio LLR generated by dec2 is deinterleaved and assisted by hard decision to obtain the final decoding output sequence. It can be seen from the above decoding process that the external information of the turbo code is fed back from another decoder, and the structure is like a turbo engine (turbo), which is the origin of the name “turbo code.”

Under the condition of additive white Gaussian noise channel, the codeword received by the decoding receiver is [21]

$$y_k^\ell = \sqrt{E_s} c_k^\ell + n_k^\ell, \ell \in \{s, p1, p2\}, \tag{1}$$

where $E_s$ represents the symbol energy of the symbol. When using BPSK modulation, $E_s = 1 \circ c_k^l$ is the encoded codeword, and the value is 1, −1. $n_k^\ell$ is Gaussian noise with zero mean variance $\sigma^2$. $y_k^\ell$ represents the codeword received by the decoding end.

The log-likelihood ratio (LLR) is defined as follows:

$$L(u_k) = \log\left(\frac{p\left(u_k = +1 | y_1^N\right)}{p\left(u_k = -1 | y_1^N\right)}\right). \tag{2}$$

Based on the decision rule of the maximum a posteriori probability MAP criterion, we can get

$$\hat{u}_k = \mathrm{sign}[L(u_k)]. \tag{3}$$

Among them, sign[.] is the sign function.

Further derivation from formula (2), we can get

$$\begin{aligned} L(u_k) = \log\left(\frac{p\left(u_k = +1 | y_1^N\right)}{p\left(u_k = -1 | y_1^N\right)}\right) &= \log\left(\frac{\sum_{U+} p\left(s_{k-1} = s', s_k = s, y_1^N\right)}{\sum_{U-} p\left(s_{k-1} = s', s_k = s, y_1^N\right)}\right) \\ &= \log\left(\frac{\sum_{U+} p\left(s', y_1^{k-1}\right) p\left(s, y_k | s'\right) p\left(y_{k+1}^{N-1} | s\right)}{\sum_{U-} p\left(s', y_1^{k-1}\right) p\left(s, y_k | s'\right) p\left(y_{k+1}^{N-1} | s\right)}\right) \\ &= \log\left(\frac{\sum_{U_+} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)}{\sum_{U-} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)}\right). \end{aligned} \tag{4}$$

It can be seen from the above formula that when calculating $L(u_k)$, it is necessary to calculate the forward state metric $\alpha_{k-1}(s')$, the branch metric $\gamma_k(s', s)$, and the backward state metric $\beta(s)$.

The forward state metric $\alpha_k(s)$ can be calculated by the following recursive calculation method:

$$\begin{aligned}\alpha_k(s) &= p\left(s, y_1^k\right) = \sum_{s'} p\left(s', s, y_1^k\right) = \sum_{s'} p\left(s', s, y_k, y_1^{k-1}\right)\\ &= \sum_{s'} p\left(s, y_k | s', y_1^{k-1}\right) p\left(s', y_1^{k-1}\right) = \sum_{s'} p\left(s, y_k | s'\right) p\left(s', y_1^{k-1}\right)\\ &= \sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s).\end{aligned} \tag{5}$$

Since $\alpha_k(s)$ is calculated recursively from forward to backward, an initial value needs to be assigned at time 0. For example, in the LTE standard, the initial state of coding is the 0 state, so the probability value of the forward state metric of the zero state at time 0 is 1, and the probability value of the forward state metric of other states is 0, which can be expressed by the following formula:

$$\alpha_0(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0. \end{cases} \tag{6}$$

The backward state metric $\beta_{k-1}(s')$ can be calculated by the following recursive calculation method:

$$\begin{aligned}\beta_{k-1}(s') &= p\left(y_k^{N-1} | s'\right) = \sum_{s} p\left(s, y_k^{N-1} | s'\right)\\ &= \sum_{s} p\left(y_{k+1}^{N-1} | s', s, y_k\right) p\left(s, y_k | s'\right)\\ &= \sum_{s} p\left(y_{k+1}^{N-1} | s\right) p\left(s, y_k | s'\right)\\ &= \sum_{s} \beta_k(s)\gamma_k(s', s).\end{aligned} \tag{7}$$

Since $\beta_{k-1}(s')$ is calculated recursively from backward to forward, an initial value needs to be assigned at the Nth moment. For example, in the LTE standard, it needs to be reset to zero after encoding. Therefore, the probability value of the backward state metric of the zero state at time $N + v$ is 1, and the probability value of the backward state metric of other states is 0, which can be expressed by the following formula:

$$\beta_{N+v}(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0. \end{cases} \tag{8}$$

The branch metric $\gamma_k(s', s)$ can be expressed as

$$\begin{aligned}\gamma_k(s', s) = p(s_k = s, y_k | s_{k-1} = s') &= \frac{p(s', s, y_k)}{p(s')} \cdot \frac{p(s', s)}{p(s', s)}\\ &= p(y_k | s', s) \cdot p(s | s')\\ &= p(y_k | u_k) \cdot p(u_k).\end{aligned} \tag{9}$$

The first term on the right side of formula (9) is the conditional probability of receiving $y_k$ under the condition that the information bit $u_k$ is known at time $k$. From formula (1), there are

$$\begin{aligned}p(y_k | u_k) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{\left\| y_k - c_k \right\|^2}{2\sigma^2}\right]\\ &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{\left(y_k^u - c_k^u\right)^2 + \left(y_k^p - c_k^p\right)^2}{2\sigma^2}\right].\end{aligned} \tag{10}$$

The second term on the right side of formula (9) is the state transition probability of the branch in the trellis graph, which is determined by the prior information $L_a(u_k)$ of the information bit $u_k$, and then, we have

$$p(u_k = -1) + p(u_k = +1) = 1, \tag{11}$$

$$L_a(u_k) = \log\left(\frac{p(u_k = +1)}{p(u_k = -1)}\right), \tag{12}$$

$p(u_k)$ can be calculated from formulas (11) and (12), as follows:

$$\begin{cases} p(u_k = +1) = \dfrac{\exp(L_a(u_k))}{1 + \exp(L_a(u_k))} \\ \\ p(u_k = -1) = \dfrac{1}{1 + \exp(L_a(u_k))} \end{cases}. \tag{13}$$

Substituting formulas (10) and (13) into formula (9), the calculation formula of branch metric can be obtained as follows:

$$\gamma_k\left(c_k^u, c_k^p\right) = \begin{cases} \dfrac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\dfrac{\left(y_k^u - c_k^u\right)^2 + \left(y_k^p - c_k^p\right)^2}{2\sigma^2}\right] \cdot \dfrac{\exp(L_a(u_k))}{1 + \exp(L_a(u_k))} & u_k = +1 \\ \\ \dfrac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\dfrac{\left(y_k^u - c_k^u\right)^2 + \left(y_k^p - c_k^p\right)^2}{2\sigma^2}\right] \cdot \dfrac{1}{1 + \exp(L_a(u_k))} & u_k = -1 \end{cases}. \tag{14}$$

If it is the first component decoding of the first iteration, let the initial value of the prior information be 0. In the subsequent component decoding, the value of the prior information is the value of the extrinsic information $L_e$ obtained by the previous component decoding after interleaving and deinterleaving. Among them, the external information $L_e(u_k)$ can be calculated by the following formula:

$$L_e(u_k) = L(u_k) - \frac{2}{\sigma^2} y_k^u - L_a(u_k). \tag{15}$$

In engineering applications, the MAP algorithm is prone to numerical instability. In order to solve this problem, it is necessary to normalize $\alpha_k(s)$ and $\beta_k(s)$ to ensure that $\alpha_k(s)$ and $\beta_k(s)$ satisfy the following formula, respectively:

$$\sum_{s=0}^{2^v-1} \alpha_k(s) = 1, \quad \sum_{s=0}^{2^v-1} \beta_k(s) = 1. \tag{16}$$

The Log-MAP decoding algorithm is also known as the MAP algorithm in the logarithmic domain. By taking the logarithm, the algorithm converts a large number of multiplication and division operations in the MAP algorithm into the corresponding addition and subtraction operations, which greatly simplifies the operation. In the Log-MAP algorithm, three new probability values are introduced as follows:

$$\begin{aligned}\widetilde{\alpha}_k(s) &= \log(\alpha_k(s))\\ \widetilde{\gamma}_k(s', s) &= \log(\gamma_k(s', s))\\ \widetilde{\beta}_k(s) &= \log(\beta_k(s)).\end{aligned} \tag{17}$$

According to the derivation in the MAP algorithm, the calculation of the forward state metric $\widetilde{\alpha}_k(s)$ in the Log-MAP algorithm can be expressed as

$$\widetilde{\alpha}_k(s) = \max{}_{s'}^{*}\left(\widetilde{\alpha}_{k-1}(s') + \widetilde{\gamma}_k(s', s)\right). \tag{18}$$

Among them, there are

$$\max{}^{*}(x, y) = \log(e^x + e^y) = \max(x, y) + \ln\left(1 + e^{-|x-y|}\right). \tag{19}$$

The calculation of the backward state metric $\widetilde{\beta}_{k-1}(s')$ can be expressed as

$$\widetilde{\beta}_{k-1}(s') = \max{}_{s}^{*}\left(\widetilde{\beta}_k(s) + \widetilde{\gamma}_k(s', s)\right). \tag{20}$$

The initial value of the state metric is

$$\widetilde{\alpha}_0(s) = \begin{cases} 0 & s = 0\\ -\infty & s \neq 0\end{cases}, \widetilde{\beta}_{N+v}(s) = \begin{cases} 0 & s = 0\\ -\infty & s \neq 0\end{cases}. \tag{21}$$

The computation of the branch metric $\widetilde{\gamma}_k(s', s)$ can be expressed as

$$\widetilde{\gamma}_k(s', s) = \begin{cases} -\dfrac{(y_k^u - c_k^u)^2 + \left(y_k^p - c_k^p\right)^2}{2\sigma^2} + L_a(u_k) - \log(1 + \exp(L_a(u_k))) & u_k = +1\\[2ex] -\dfrac{(y_k^u - c_k^u)^2 + \left(y_k^p - c_k^p\right)^2}{2\sigma^2} - \log(1 + \exp(L_a(u_k))) & u_k = -1\end{cases}. \tag{22}$$

The calculation of the log-likelihood ratio $L(u_k)$ can be expressed as

$$\begin{aligned}L(u_k) = {}&\max{}_{U+}^{*}\left[\widetilde{\alpha}_{k-1}(\dot{s}') + \widetilde{\gamma}_k(s', s) + \widetilde{\beta}_k(s)\right]\\ &- \max{}_{U-}^{*}\left[\widetilde{\alpha}_{k-1}(s') + \widetilde{\gamma}_k(s', s) + \widetilde{\beta}_k(s)\right].\end{aligned} \tag{23}$$

The calculation of external information $L_e(u_k)$ can be expressed as

$$L_e(u_k) = L(u_k) - \frac{2}{\sigma^2} y_k^u - L_a(u_k). \tag{24}$$

There are a large number of $\max^*(.)$ operations in the Log-MAP algorithm, which includes exponential and logarithmic operations and still has a very large computational complexity. In order to further simplify the operation, the logarithmic operation $\ln(1 + e^{-|x-y|})$ term in the $\max^*(.)$ operation is ignored, which is called the Max-Log-MAP decoding algorithm. In the Max-Log-MAP algorithm, there are only addition and subtraction operations. Compared with the Log-MAP algorithm, the decoding computational complexity is greatly simplified, but the performance is poor.

Generally speaking, ($n$, $p$) fixed-point processing includes three steps: absolute quantization, fixed decimal places, and fixed integer bits. Absolute quantization is to convert the real number $Y$ into an integer $Y_i$ for subsequent determination of binary decimal places and binary integer bits. The specific process is as follows:

$$y_i = \lfloor 2^P\colon\ y + 0.5 \rfloor. \tag{25}$$

Among them, $L$ means round down. The processing of fixed decimal places is to shift the integer $Q$ to the right by $p$ bits to obtain a real number $y_q$ including $p$ binary decimal places; that is,

$$y_q = \frac{y_i}{2^P}. \tag{26}$$

The number of decimal places in the fixed-point data is related to the minimum resolution of the fixed-point data. The choice of the fixed-point decimal place width and its impact on the performance of bit error rate and frame error rate will be discussed in the performance simulation of this chapter. Limiting is to limit the real number $y_q$ that exceeds the range of (n, p) fixed-point representation, and its calculation method is

$$y_l = \begin{cases} -2^{n-p-1} + 2^{-p} & y_q \leq -2^{n-p-1} \\ y_q & -2^{n-p-1} < y_q < 2^{n-p-1}. \\ 2^{n-p-1} - 2^{-p} & y_q \geq 2^{n-p-1} \end{cases} \tag{27}$$

After the fixed-point decimal bit width P is determined, the following will analyze the fixed-point integer bit width of the data to be decoded, a priori information/external information, branch metrics and state metrics, so as to determine the fixed-point mode.

We assume that under AWGN channel conditions, the BPSK debugging method is adopted, and the coded symbols received at time $k$ can be expressed as

$$y_k^l = c_k^l + n_k^l, \ell \in \{s, p1, p2\}, \tag{28}$$

where $y_k^\ell$ is the data received by the decoding end, $c_k^\ell \in \{1, -1\}$ represents the coded symbol after BPSK modulation, and $n_k^\ell$ is a Gaussian noise whose mean variance is $\sigma^2$.

First, the fixed-point bit width of the data $y_k^\ell$ to be decoded needs to be determined. When selecting $(n, p)$ fixed-point processing, the following probability $y_k^\ell$ should be made small enough; that is,

$$\Pr\left\{y_k^\ell \leq -2^{n-p-1} + 2^{-p}\right\} + \Pr\left\{y_k^\ell \geq 2^{n-p-1} - 2^{-p}\right\} = \varepsilon. \tag{29}$$

Since the receiver does not know the codeword information of the sender, it is possible to make the sender's codeword equal to 1 and −1; that is,

$$\Pr\left\{c_k^l = 1\right\} = \Pr\left\{c_k^l = -1\right\} = \frac{1}{2}. \tag{30}$$

Therefore, there are

$$\begin{aligned} &\Pr\left\{y_k^l \leq -2^{n-p-1} + 2^{-p}\right\} \\ &= \Pr\left\{y_k^l \leq -2^{n-p-1} + 2^{-p}, c_k' = 1\right\} + \Pr\left\{y_k' \leq -2^{n-p-1} + 2^{-p}, c_k' = -1\right\} \\ &= \Pr\left\{y_k^l \leq -2^{n-p-1} + 2^{-p} | c_k^l = 1\right\} \Pr\left\{c_k^l = 1\right\} + \Pr\left\{y_k' \leq -2^{n-p-1} + 2^{-p} | c_k^l = -1\right\} \Pr\left\{c_k^l = -1\right\} \\ &= \frac{1}{2} Q\left(\frac{2^{n-p-1} - 2^{-p} + 1}{\sigma}\right) + \frac{1}{2} Q\left(\frac{2^{n-p-1} - 2^{-p} - 1}{\sigma}\right). \end{aligned} \tag{31}$$

In the same way, we can get

$$\begin{aligned} \Pr\left\{y_k' \geq 2^{n-p-1} - 2^{-p}\right\} = \frac{1}{2} Q\left(\frac{2^{n-p-1} - 2^{-p} - 1}{\sigma}\right) \\ + \frac{1}{2} Q\left(\frac{2^{n-p-1} - 2^{-p} + 1}{\sigma}\right). \end{aligned} \tag{32}$$

The $Q$-function is monotonically decreasing and decreases rapidly. In order to simplify the formula, the following two approximations are made: the one is that $2^{n-p-1} - 2^{-p} \approx 2^{n-p-1}$ and the other is that the term of $Q(2^{n-p-1} - 2^{-p} + 1/\sigma)$ in formula (31) is ignored. Substituting formulas (30) and (31) into formula (29), we get

$$\Pr\left\{y_k' \leq -2^{n-p-1} + 2^{-p}\right\} + \Pr\left\{y_k' \geq 2^{n-p-1} - 2^{-p}\right\} = Q\left(\frac{2^{n-p-1} - 1}{\sigma}\right) = \varepsilon. \tag{33}$$

According to the above formula, it can be estimated that the fixed-point integer bit width $(n - p - 1)$ required for the data to be decoded under the condition that the specific overflow probability is $\varepsilon$ is

$$\begin{aligned} n - p - 1 &= \left\lceil \log_2\left\{1 + Q^{-1}(\varepsilon) \cdot \sigma\right\}\right\rceil \\ &= \left\lceil \log_2\left\{1 + Q^{-1}(\varepsilon) \cdot \sqrt{\frac{N_o}{R_c E_b}}\right\}\right\rceil, \end{aligned} \tag{34}$$

where ⌈⌉ represents rounding up, and $Q^{-1}(\cdot)$ represents the inverse function of the $Q(\cdot)$ function, which can usually be determined by looking up the table. Obviously, for a specific coding scheme and a set signal-to-noise ratio $(E_b/\sigma^2)_{dB}$ interval (often corresponding to a specific reliability performance requirement), the fixed-point integer bit width required for the data to be decoded in the fixed-point decoding algorithm can be calculated by formula (34).

Then, as the prior information for subsequent component decoding, the prior information also approximately obeys the Gaussian distribution; that is, $L_a \propto N(\mu_{L_a}, \sigma_{L_a}^2)$, and $\mu_{L_a} = \mu_{L_e}, \sigma_{L_a}^2 = \sigma_{L_e}^2$. Similar analysis methods can be used to determine the fixed-point way of extrinsic information and prior information. The fixed-point method for analyzing prior information satisfies the following relationship:

$$\Pr\left\{L_a \le -2^{n_{L_e}-p-1} + 2^{-p}\right\} + \Pr\left\{L_a \ge 2^{n_{L_0}-p-1} - 2^{-p}\right\} = \varepsilon$$
$$\Rightarrow Q\left(\frac{2^{n_{b_b}-p-1} - 2^{-p} + \mu_{L_o}}{\sigma_{L_e}}\right) + Q\left(\frac{2^{n_{L_b-p-1}} - 2^{-p} - \mu_{L_e}}{\sigma_{L_e}}\right) = \varepsilon \tag{35}$$
$$\Rightarrow Q\left(\frac{2^{n_{b_e}-p-1} - \mu_{L_e}}{\sigma_{L_e}}\right) = \varepsilon.$$

According to the above formula, it can be estimated that under the condition of a specific overflow probability $\varepsilon$, the fixed-point integer bit width $(n_{L_e} - p - 1)$ required by the prior information and the extrinsic information is

$$n_{L_a} - p - 1 = \left[\log_2\left(\mu_{L_a} + Q^{-1}(\varepsilon)\sigma_{L_e}\right)\right]. \tag{36}$$

Among them, the calculation methods of $\mu_{L_a}$ and $\sigma^2_{L_a}$ will be introduced in detail later.

The branch metric data bit width is determined.

In the fixed-point decoding algorithm, formula (22) can be simplified as the following formula:

$$\gamma_k(s', s) = y_k^s \cdot z_k^s + y_k^p \cdot z_k^p + \frac{1}{2} \cdot z_k^s \cdot L_{a,k}. \tag{37}$$

Substituting formula (28) into the above formula, we can get

$$\gamma_k(s', s) = z_k^s \cdot c_k^s + z_k^p \cdot c_k^p + z_k^s \cdot n_k^s + z_k^p \cdot n_k^p + \frac{1}{2} \cdot z_k^s \cdot L_{a,k}. \tag{38}$$

Among them, $z_k^s, z_k^p, c_k^s, c_k^p \in \{1, -1\}, n_k^s, n_k^p \sim N(0, \sigma^2)$, $L_{a,k} \sim N(\mu_{L_a}, \sigma^2_{L_a})$, it can be obtained that the branch metric obeys the Gaussian distribution, as shown in the following formula:

$$\gamma_k(s', s) \propto N\left(z_k^s \cdot c_k^s + z_k^p \cdot c_k^p + \frac{1}{2} z_k^s \cdot \mu_{L_e}, 2\sigma^2 + \frac{1}{4}\sigma^2_{L_a}\right). \tag{39}$$

It can be seen from the above formula that if the branch metric is fixed-point processing, the overflow probability $\varepsilon$ should be made small enough; that is,

$$\Pr\left\{\gamma_k \le -2^{n-p-1} + 2^{-p}\right\} + \Pr\left\{\gamma_k \ge 2^{n-p-1} - 2^{-p}\right\} = \varepsilon. \tag{40}$$

Different values of $(z_k^s, z_k^p)$ and its relationship with $(c_k^s, c_k^p)$ need to be considered in the calculation of the analysis branch metric. $\gamma_k(s', s)$ can also be represented by $\gamma_k(z_k^s, z_k^p)$, and the corresponding probability can be approximated as

$$\begin{aligned}
\Pr\left\{\gamma_k\left(z_k^k, z_k^p\right) \le -2^{n-p-1} + 2^{-p}\right\} &= \Pr\left\{\gamma_k\left(z_k^s, z_k^p\right) \le -2^{n-p-1} + 2^{-p} | z_k^s = 1, z_k^p = 1\right\} \cdot \Pr\left\{z_k^s = 1, z_k^p = 1\right\} \\
&\quad + \Pr\left\{\gamma_k\left(z_k^s, z_k^p\right) \le -2^{n-p-1} + 2^{-p} | z_k^s = -1, z_k^p = 1\right\} \cdot \Pr\left\{z_k^s = -1, z_k^p = 1\right\} \\
&\quad + \Pr\left\{\gamma_k\left(z_k^s, z_k^p\right) \le -2^{n-p-1} + 2^{-p} | z_k^s = -1, z_k^p = -1\right\} \cdot \Pr\left\{z_k^s = -1, z_k^p = -1\right\} \\
&\quad + \Pr\left\{\gamma_k\left(z_k^s, z_k^p\right) \le -2^{n-p-1} + 2^{-p} | z_k^s = 1, z_k^p = -1\right\} \cdot \Pr\left\{z_k^s = 1, z_k^p = -1\right\} \\
&= \frac{1}{4}\left[\begin{array}{c} Q\left(\dfrac{2^{n-p-1} - 2^{-p} + \left(x_k^s + x_k^p + 0.5\mu_{L_o}\right)}{\sqrt{2\sigma^2 + 0.25\sigma^2_{L_0}}}\right) + Q\left(\dfrac{2^{n-p-1} - 2^{-p} + \left(x_k^s - x_k^p + 0.5\mu_{L_o}\right)}{\sqrt{2\sigma^2 + 0.25\sigma^2_{L_e}}}\right) \\ + Q\left(\dfrac{2^{n-p-1} - 2^{-p} + \left(-x_k^2 + x_k^p - 0.5\mu_{L_0}\right)}{\sqrt{2\sigma^2 + 0.25\sigma^2_{L_0}}}\right) + Q\left(\dfrac{2^{n-p-1} - 2^{-p} + \left(-x_k^s - x_k^p - 0.5\mu_{L_k}\right)}{\sqrt{2\sigma^2 + 0.25\sigma^2_{L_0}}}\right) \end{array}\right] \\
&\approx \frac{1}{4} Q\left(\frac{2^{n-p-1} - \left(\left|x_k^2 + 0.5\mu_{L_e}\right| + \left|x_k^p\right|\right)}{\sqrt{2\sigma^2 + 0.25\sigma^2_{L_k}}}\right) \approx \frac{1}{4} Q\left(\frac{2^{n-p-1} - \left(\left|x_k^2\right| + \left|0.5\mu_{L_0}\right| + \left|x_k^p\right|\right)}{\sqrt{2\sigma^2 + 0.25\sigma^2_{L_0}}}\right) \\
&= \frac{1}{4} Q\left(\frac{2^{n-\rho-1} - \left(2 + 0.5\mu_{L_e}\right)}{\sqrt{2\sigma^2 + 0.25\sigma^2_{L_e}}}\right).
\end{aligned} \tag{41}$$

In the same way, we can get

$$\Pr\left\{\gamma_k\left(z_k^s, z_k^p\right) \geq 2^{n-p-1} - 2^{-p}\right\} = \frac{1}{4}Q\left(\frac{2^{n-p-1} - \left(2 + 0.5\mu_{L_e}\right)}{\sqrt{2\sigma^2 + 0.25\sigma_{L_e}^2}}\right). \quad (42)$$

Substitute formulas (41) and (42) into formula (40), we can get

$$\Pr\left\{\gamma_k \leq -2^{n-p-1} + 2^{-p}\right\} + \Pr\left\{\gamma_k \geq 2^{n-p-1} - 2^{-p}\right\} = \frac{1}{2}Q\left(\frac{2^{n-p-1} - \left(2 + 0.5\mu_{L_o}\right)}{\sqrt{2\sigma^2 + 0.25\sigma_{L_o}^2}}\right) = \varepsilon. \quad (43)$$

According to the above formula, it can be estimated that under the condition of a specific overflow probability $\varepsilon$, the fixed-point integer bit width $(n_\gamma - p - 1)$ required by the branch metric is

$$n_\gamma - p - 1 = \left[\log_2\left\{2 + 0.5\mu_{L_d} + Q^{-1}(2\varepsilon)\cdot\sqrt{2\sigma^2 + 0.25\sigma_{L_a}^2}\right\}\right]. \quad (44)$$

The state metric data bit width is determined.

According to formulas (18) and (20), the state metric is obtained by adding two numbers, which must satisfy $n_s > n_\gamma$, where $n_s$ represents the fixed-point total bit width of the state metric. Since the state metric $\alpha$ (or $\beta$) is calculated recursively, its value will continue to increase during the recursive calculation process. In order to prevent fixed-point overflow in the recursive process, $\alpha$ (or $\beta$) needs to be normalized. The selection and processing method of normalization will be analyzed in detail in the next section, and the specific integer bit width of the state metric will be determined.

In floating-point computing, the commonly used state metric normalization methods are as follows:

$$\widetilde{\alpha}_k(s) = \alpha_k(s) - \max_{s'}\left[\alpha_k(s')\right], \widetilde{\beta}_k(s) = \beta_k(s) - \max_{s'}\left[\beta_k(s')\right]. \quad (45)$$

However, the normalization method of the above formula will bring additional calculation delay, and there is no guarantee that the calculation overflow will not occur when the data bit width is limited. Therefore, it cannot be directly extended to the fixed-point decoding algorithm.

In the recursive calculation of state metrics in turbo decoding, a feasible normalization scheme is to judge the values of all state metrics $\alpha_k(s')$ and $\beta_{k+1}(s)$ at the previous moment before recursive calculation. Normalization is performed on recursive computations in cases where computation overflow is possible. The above normalization scheme can be expressed by the following formulas:

$$\alpha_{k+1}(s) = \begin{cases} \max_{s'}\left[\alpha_k(s') + \gamma_{k+1}(s', s)\right] & \text{if } \forall s', \alpha_k(s') < \alpha_{\max} - \omega \\ \max_{s'}\left[\alpha_k(s') + \gamma_{k+1}(s', s) - \omega\right] & \text{if } \exists s', \alpha_k(s') \geq \alpha_{\max} - \omega \end{cases}, \quad (46)$$

$$\beta_k(s') = \begin{cases} \max_s\left[\beta_{k+1}(s) + \gamma_{k+1}(s', s)\right] & \text{if } \forall s, \beta_{k+1}(s) < \beta_{\max} - \omega \\ \max_s\left[\beta_{k+1}(s) + \gamma_{k+1}(s', s) - \omega\right] & \text{if } \exists s, \beta_{k+1}(s) \geq \beta_{\max} - \omega. \end{cases}, \quad (47)$$

where $\alpha_{\max}$ and $\beta_{\max}$ are the maximum values of the state metric in the fixed-point mode. If the fixed-point mode of the state metric is $(n_s, p)$, then $\alpha_{\max} = \beta_{\max} = 2^{n_s - p - 1} - 2^{-p}$. $\omega$ is the preset normalization constant during normalization processing. A simple method is to measure the maximum value of the fixed-point $(n_\gamma, p)$ manner according to the aforementioned branch metric; that is, $\omega = \gamma_{\max}$. In this way, if some values of the state metrics $\alpha_k(s')$ and $\beta_{k+1}(s)$ at the previous moment exceed the decision threshold $\alpha_{\max} - \omega$ or $\beta_{\max} - \omega$, the aforementioned normalization operation is performed, thereby ensuring that the calculation of the state metrics will not occur calculation overflow. The normalized threshold value $\omega$ is required, and the branch metric, a priori information, and the fixed-point mode of the data to be decoded are calculated.

Based on random statistical analysis, an effective method for calculating the Gaussian distribution of prior information/extrinsic information in the iterative process is presented. The calculation formula is as follows:

$$L_e^t = -\ln\left(\exp\left(-L_+\right) + \exp\left(-L_-\right)\right), \quad (48)$$

where the variables $L_+$ and $L_-$ can be calculated iteratively as follows:

$$\begin{aligned} L_\pm &= \rho_\pm + L_\pm^{(1)} \\ L_\pm^{(i)} &= \delta_\pm^{(i)} - \ln\left(\exp\left(-\eta_\pm^{(i)}\right) + \exp\left(-L_\pm^{(i+1)}\right)\right) \quad i = 1, 2, \ldots, M-1 \\ L_\pm^{(M)} &= \delta_\pm^{(M)} + \eta_\pm^{(M)}, \end{aligned} \quad (49)$$

where $\rho_\pm, \delta_\pm^{(i)}, \eta_\pm^{(i)}$ are independent Gaussian distributed random variables, and their probability distribution characteristics are as follows:

$$\begin{aligned} \rho_\pm &\sim N\left(\mu_r, \sigma_r^2\right) \\ \delta_\pm^{(i)} &\sim N\left(\left(w_0 - 2\right)\mu_r, \left(w_0 - 2\right)\sigma_r^2\right) \\ \eta_\pm^{(i)} &\sim N\left(2\mu_r + \mu_a, 2\sigma_r^2 + \sigma_a^2\right), \end{aligned} \quad (50)$$

where $\mu_r$ and $\sigma_r^2$ are the mean and variance of the data $y_k^\ell$ to be decoded. $\mu_a$ and $\sigma_a^2$ are the mean sum and variance of the current iterative decoding prior information. When $t = 1$, the initial value of the prior information is 0; that is, $\mu_a = \sigma_a^2 = 0$. When $t > 1$, the mean and variance of the prior information are set to the mean and variance of the external information in $(t - 1)$ iterations of decoding, respectively. $M$ is the number of sequences with a code weight of 2. In the paper, $M = 2, w_0$ is the minimum codeword weight corresponding to the input sequence of code weight 2, when the component encoder is (1, 15/13), $w_0 = 6$.

$$Z = \ln\left(\exp\left(X_1\right) + \exp\left(X_2\right)\right), X_1 - N\left(\mu_1, \sigma_1^2\right), X_2 \sim N\left(\mu_2, \sigma_2^2\right), \tag{51}$$

$$Z - N\left(\mu_z, \sigma_z^2\right)\begin{cases} \mu_Z = 2\ln m_1 - \dfrac{1}{2}\ln m_2 \\ \sigma_Z^2 = \ln m_2 - 2\ln m_1 \end{cases}. \tag{52}$$

Among them, there are

$$\begin{aligned} \varepsilon\left[\exp\left(X_i\right)\right] &= \exp\left(\frac{2\mu_i + \sigma_i^2}{2}\right) \quad i = 1, 2 \\ m_1 &= \varepsilon\left[\exp\left(Z\right)\right]\sum_{i=1}^{2}\exp\left(\frac{2\mu_i + \sigma_i^2}{2}\right) \\ m_2 &= \varepsilon\left[\left(\exp\left(Z\right)\right)^2\right] = \sum_{i=1}^{n}\exp\left(2\mu_i + 2\sigma_i^2\right) + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n}\exp\left(\mu_i + \mu_j + \frac{\sigma_i^2 + \sigma_j^2 + 2r_{ij}\sigma_i\sigma_j}{2}\right) \\ &= \exp\left(2\mu_1 + 2\sigma_1^2\right) + \exp\left(2\mu_2 + 2\sigma_2^2\right) + 2\exp\left(\mu_1 + \mu_2 + \frac{\sigma_1^2 + \sigma_2^2}{2}\right). \end{aligned} \tag{53}$$

The logarithmic distributions of the exponential sums involved in formulas (48) and (49) can be approximated by the Fenton–Wilkinson (FW) method, as shown in formulas (51) and (52). The mean and variance of the outer information $L_e^t/a$ priori information $L_a^{t+1}$ of the $t$-th iteration can be calculated by formulas (48), (49), and (52).

## 4. Path Selection Strategy of Communication Network Based on Graph Convolutional Neural Network

In the underlying architecture of the system, a set of flow control group policies is implemented through the SDN-based network flow control module, so that the system can adjust the network state according to the analysis results of the network situation information, and control and migrate the flow in the entire network. The system architecture diagram is shown in Figure 2.

The network architecture diagram of the scheme described in this paper is shown in Figure 3. In the entire network, the core gateway is used to control traffic and access to other subnets and also acts as a reverse proxy when the external network accesses the internal network server.

The technical scheme of the flow control module implemented based on the SDN idea adopted in this system is shown in Figure 4. In this technical solution, the modules can be divided into three layers, namely the application layer, the control layer, and the data forwarding layer.

The depth of the graph neural network in this paper is set to 2 layers. The calculation process of the graph neural network is shown in Figure 5. After each layer of network outputs the sample feature matrix, the matrix is reused to calculate its adjacency matrix; that is, the graph structure is updated. Finally, the feature vector of the sample to be classified in the graph structure is taken out, and the softmax function is used to map it to the final classification result.

Figure 6 shows the network structure of the denoising convolutional autoencoder. The amount of feature map parameters decreases layer by layer during the encoding process of the convolutional autoencoder. In order to prevent the loss of too much useful information in the encoding process, skip connections are added between the corresponding encoding and decoding layers when constructing the denoising convolutional autoencoder. The skip connection directly introduces the information of the coding layer into the corresponding deep decoding layer, and the model selects important noise information by adjusting the parameter weights, thereby reconstructing the noise of the signal.

This paper takes the IEEE802.16j/m series of standards and drafts as the platform and takes the multihop relay cellular network based on OFMDA technology as the research object to study its path selection algorithm. In addition, a system performance simulation platform of multihop relay cellular network is built, and the corresponding performance analysis of the proposed path selection algorithm is carried out. The typical multihop relay cellular network scenario based on IEEE802.16j/m series standards is shown in Figure 7.

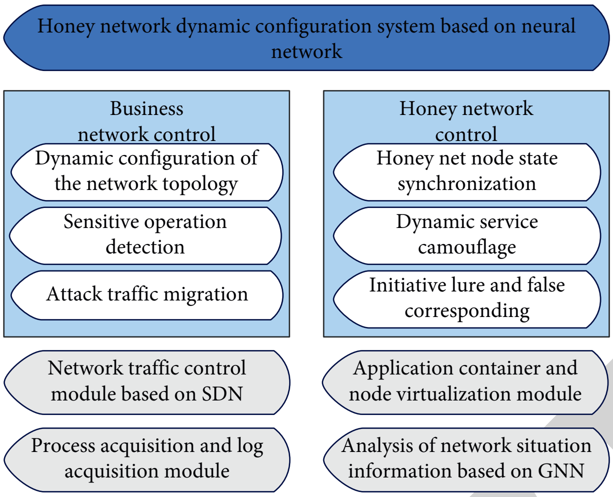
Honey network dynamic configuration system based on neural network
Business network control
Dynamic configuration of the network topology
Sensitive operation detection
Attack traffic migration
Honey network control
Honey net node state synchronization
Dynamic service camouflage
Initiative lure and false corresponding
Network traffic control module based on SDN
Application container and node virtualization module
Process acquisition and log acquisition module
Analysis of network situation information based on GNN

FIGURE 2: System architecture diagram.

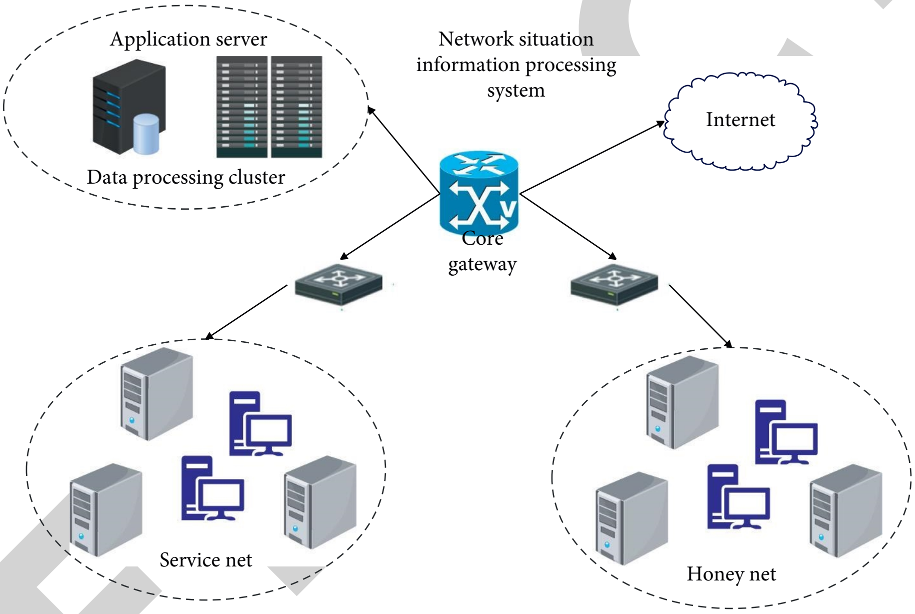

FIGURE 3: Network architecture diagram.

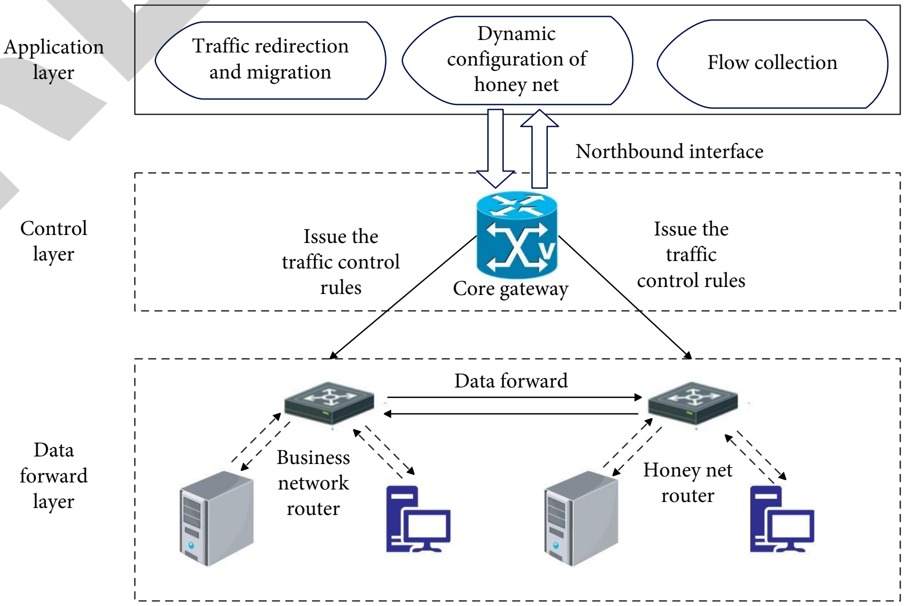

FIGURE 4: Technical scheme of flow control module.

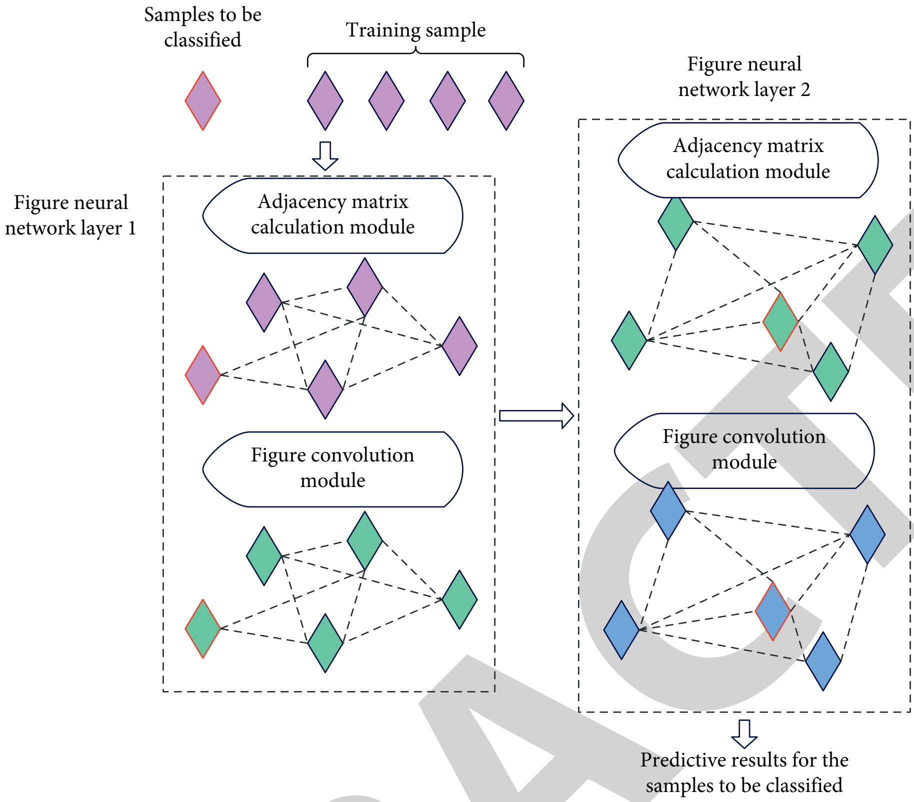
Samples to be classified
Training sample
Figure neural network layer 1
Adjacency matrix calculation module
Figure convolution module
Figure neural network layer 2
Adjacency matrix calculation module
Figure convolution module
Predictive results for the samples to be classified

Figure 5: The calculation flow of the neural network.

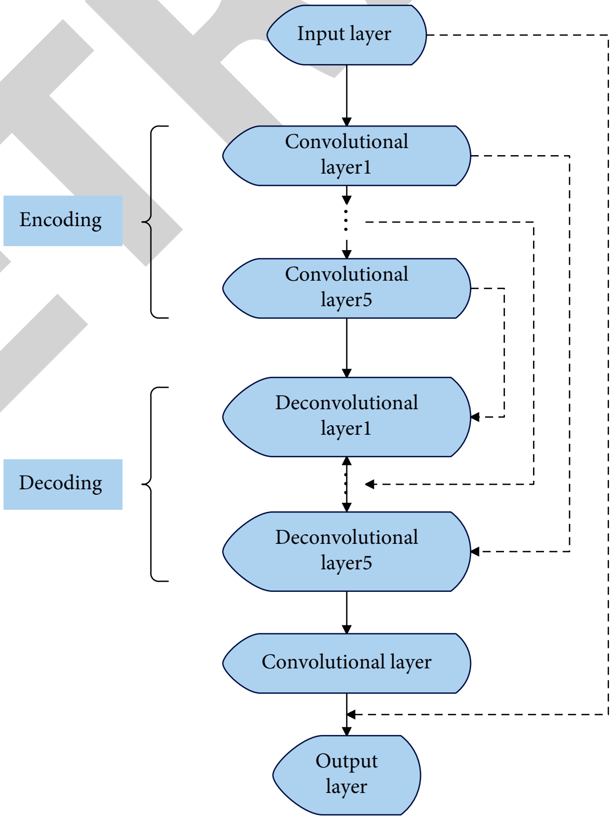

Figure 6: Network structure of denoising convolutional autoencoder.

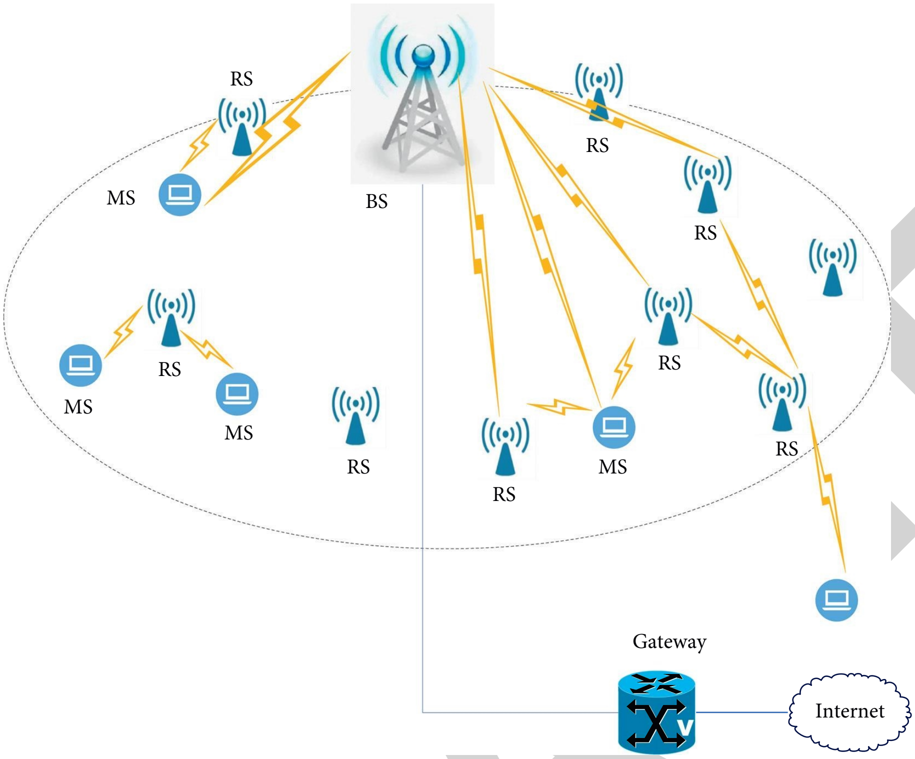

Figure 7: Multihop relay cellular network road scenario.

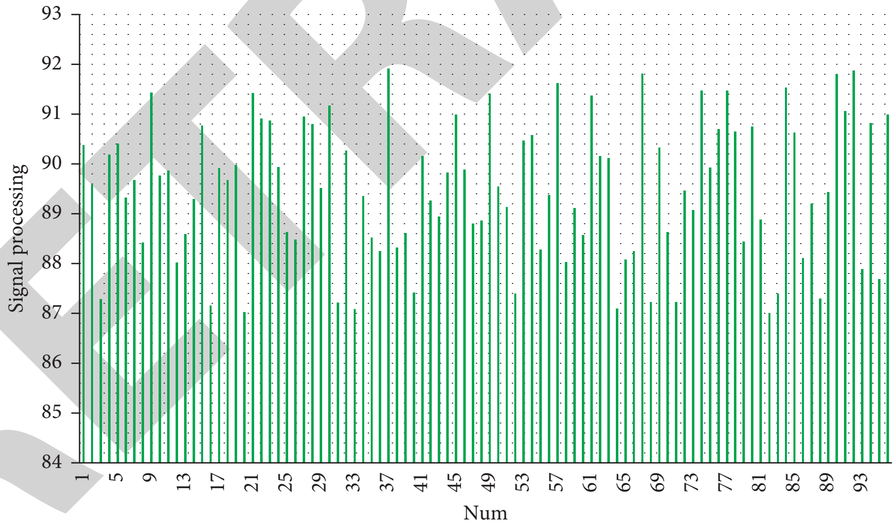

Figure 8: Signal processing effect of communication network path selection system based on graph convolutional neural network.

Based on the scene set in Figure 7, the system model of this paper is constructed and simulated. Moreover, this paper explores the path selection effect of communication network based on graph convolutional neural network. The model is simulated by Matlab, the effect of signal processing and communication path selection is counted in this paper, and the results shown in Figures 8 and 9 are obtained.

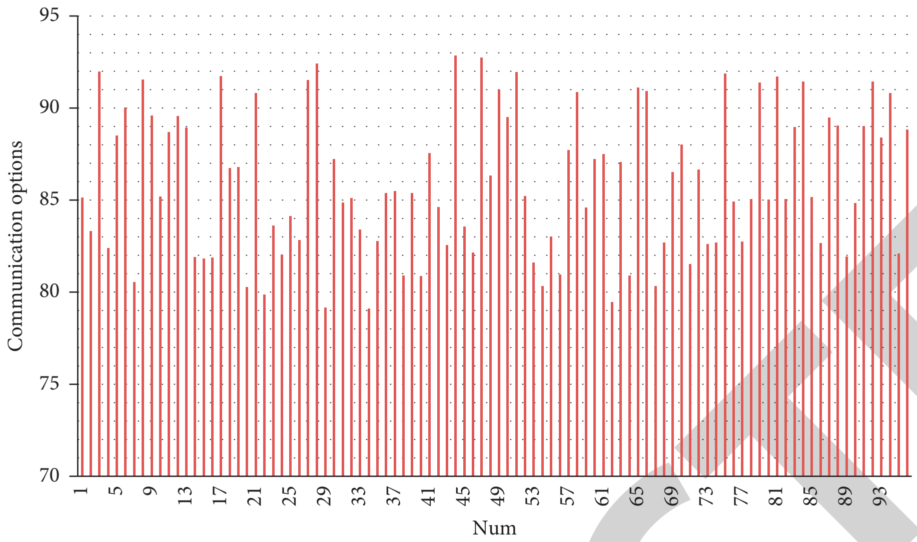

Figure 9: Path selection effect of communication network path selection system based on graph convolutional neural network.

It can be seen from the above research that the communication network path selection system based on graph convolutional neural network proposed in this paper can effectively improve the effect of multicommunication path selection.

## 5. Conclusion

The absence of a complete link between communicating nodes does not mean that the pair of nodes cannot communicate. Opportunistic routing can select the node closest to the target node to forward data, and adopts the “store-carry-forward” communication strategy. By utilizing the movement of nodes, the nodes can be brought into the mutual communication range to realize communication. This opportunistic routing mechanism can improve the efficiency, throughput and reliability of the network. A graph neural network is a mobile self-organizing network that uses an opportunistic routing mechanism. Moreover, the feature of opportunistic networks that do not require full network connectivity is more in line with the actual needs of ad hoc networks, so it has become a research hotspot in many fields in recent years. This paper combines the graph convolutional neural network to formulate the communication network path selection strategy and constructs an intelligent system. The research shows that the communication network path selection system based on graph convolutional neural network proposed in this paper can effectively improve the effect of multicommunication path selection.

## Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

## Conflicts of Interest

The author declares no conflicts of interest.

## Acknowledgments

This study was sponsored by Henan Finance University.

## References

[1] J. Pan, Q. Xie, H. Chiang et al., ““From the nature for the nature”: an eco-friendly antifouling coating consisting of poly(lactic acid)-based polyurethane and natural antifoulant,” *ACS Sustainable Chemistry & Engineering*, vol. 8, no. 3, pp. 1671–1678, 2019.

[2] B. Behroozpour, P. A. M. Sandborn, M. C. Wu, and B. E. Boser, “Lidar system Architectures and circuits,” *IEEE Communications Magazine*, vol. 55, no. 10, pp. 135–142, 2017.

[3] J. Barowski, M. Zimmermanns, and I. Rolfes, “Millimeter-Wave characterization of dielectric materials using calibrated FMCW transceivers,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 8, pp. 3683–3689, 2018.

[4] Y. Jiang, S. Karpf, and B. Jalali, “Time-stretch LiDAR as a spectrally scanned time-of-flight ranging camera,” *Nature Photonics*, vol. 14, no. 1, pp. 14–18, 2020.

[5] L. J. Xu, X. Lin, Q. He, M. Worku, and B. Ma, “Highly efficient eco-friendly X-ray scintillators based on an organic manganese halide[J],” *Nature Communications*, vol. 11, no. 1, pp. 1–7, 2020.

[6] Q. Y. Cheng, X. L. Zhao, Y. X. Weng, Y. D. Li, and J. B. Zeng, “Fully sustainable, nanoparticle-free, fluorine-free, and robust superhydrophobic cotton fabric fabricated via an eco-friendly method for efficient oil/water separation,” *ACS Sustainable Chemistry & Engineering*, vol. 7, no. 18, pp. 15696–15705, 2019.

[7] N. Maring, P. Farrera, K. Kutluer, M. Mazzera, G. Heinze, and H. de Riedmatten, “Photonic quantum state transfer between a cold atomic gas and a crystal,” *Nature*, vol. 551, no. 7681, pp. 485–488, 2017.

[8] H. Mohapatra and A. K. Rath, “Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT,” *IET Wireless Sensor Systems*, vol. 9, no. 6, pp. 447–457, 2019.

[9] A. Seri, G. Corrielli, D. Lago-Rivera et al., “Laser-written integrated platform for quantum storage of heralded single photons,” *Optica*, vol. 5, no. 8, p. 934, 2018.

[10] Z. Meng, J. Li, C. Yin et al., “Dual-band dechirping LFMCW radar receiver with high image rejection using microwave photonic I/Q mixer,” *Optics Express*, vol. 25, no. 18, p. 22055, 2017.

[11] A. Al-Halafi and B. Shihada, “UHD video transmission over bidirectional underwater wireless optical communication,” *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–14, 2018.

[12] H. C. Kumawat and A. Bazil Raj, “Extraction of Doppler signature of micro-to-macro rotations/motions using continuous wave radar-assisted measurement system,” *IET Science, Measurement & Technology*, vol. 14, no. 7, pp. 772–785, 2020.

[13] F. Zhang, Q. Guo, and S. Pan, “Photonics-based real-time ultra-high-range-resolution radar with broadband signal generation and processing,” *Scientific Reports*, vol. 7, no. 1, p. 13848, 2017.

[14] T. Zhong and P. Goldner, “Emerging rare-earth doped material platforms for quantum nanophotonics,” *Nanophotonics*, vol. 8, no. 11, pp. 2003–2015, 2019.

[15] Z. Sabouri, A. Akbari, H. A. Hosseini, A. Hashemzadeh, and M. Darroudi, “Eco-Friendly biosynthesis of nickel oxide nanoparticles mediated by okra plant extract and investigation of their photocatalytic, magnetic, cytotoxicity, and antibacterial properties,” *Journal of Cluster Science*, vol. 30, no. 6, pp. 1425–1434, 2019.

[16] T. Zhong, J. M. Kindem, J. G. Bartholomew et al., “Nanophotonic rare-earth quantum memory with optically controlled retrieval,” *Science*, vol. 357, no. 6358, pp. 1392–1395, 2017.

[17] K. Soga and L. Luo, “Distributed fiber optics sensors for civil engineering infrastructure sensing,” *Journal of Structural Integrity and Maintenance*, vol. 3, no. 1, pp. 1–21, 2018.

[18] Y. J. Ma, A. Tadros, J. Du, and E. Y. Chang, “Quantitative two-dimensional ultrashort echo time magnetization transfer (2D UTE-MT) imaging of cortical bone,” *Magnetic Resonance in Medicine*, vol. 79, no. 4, pp. 1941–1949, 2018.

[19] S. Sharaf and M. E. El-Naggar, “Eco-friendly technology for preparation, characterization and promotion of honey bee propolis extract loaded cellulose acetate nanofibers in medical domains,” *Cellulose*, vol. 25, no. 9, pp. 5195–5204, 2018.

[20] C. Zhang, P. Xiao, F. Ni et al., “Converting pomelo peel into eco-friendly and low-consumption photothermic biomass sponge toward multifunctioal solar-to-heat conversion,” *ACS Sustainable Chemistry & Engineering*, vol. 8, no. 13, pp. 5328–5337, 2020.

[21] J. Yang, T. Yang, Z. Wang, D. Jia, and C. Ge, “A novel method of measuring instantaneous frequency of an ultrafast frequency modulated continuous-wave laser,” *Sensors*, vol. 20, no. 14, p. 3834, 2020.