

Research Article

A Method of Optimizing Network Topology Structure Combining Viterbi Algorithm and Bayesian Algorithm

Xiaoxiao Shi 

Hubei University of Technology, WuHan, Hubei 430068, China

Correspondence should be addressed to Xiaoxiao Shi; suixing3693@163.com

Received 16 February 2021; Revised 23 March 2021; Accepted 11 April 2021; Published 10 May 2021

Academic Editor: Rahim Khan

Copyright © 2021 Xiaoxiao Shi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With Internet entering all walks of life, development of internet and usage expansion demand better performance, especially the application of 5G network that adopts NAS networking mode. Some of the network bandwidth cannot fully support the current network demand, which causes network fluctuations and other concerns. In this paper, a method for optimizing the topological structure of the bottom layer of the communication network is proposed that has outage performance close to optimal routing scheme. In specific, path in areas with poor network conditions is first optimized using Viterbi algorithm. Then, network element nodes on the path are optimized using Bayes recommendation algorithm for reasonable flow distribution. Dual planning of improved Viterbi algorithm is used to realize the main and standby path planning, and then, Bayesian recommendation algorithm based on the average value is used to optimize the network elements. Therefore, it is very efficient to realize overall performance optimization.

1. Introduction

Network topology refers to the physical layout of interconnecting various devices with transmission media, that is, how to connect computers and other devices in the network. The topology diagram shows the network configuration and mutual connection of network servers and workstations. Its structure mainly includes star structure, ring structure, bus structure, distributed structure, tree structure, mesh structure, and honeycomb structure. In the structural hierarchy, most networks adopt a three-layer network architecture: access layer, convergence layer, and core layer. The stable operation of the overall network is achieved through the rational distribution of the three-tier structure.

Based on the physical layout, there are also some logical structure designs. The logical structure design makes the data transmission more efficient and stable. There are some problems with the current logical network structure, such as instability, the design of the logical structure does not consider the whole, and less consideration for scalability. And due to the development of 5G networks, there are higher requirements on the current network conditions for 5G networks that

adopt nonindependent networking (NSA) mode [1]. Since it is not suitable for redesigning the entire network topology in nonindependent networking, partial gradual optimization is required to make the entire network system more stable and efficient.

In the overall network structure, the topological structure of the communication network consists of two parts, namely, the network element node, which is actually a single transit base station, which receives, processes, and forwards all communication data. Among the forwarded network elements, the board is involved. With ports, there are multiple boards in the same network element, and there are multiple ports to choose from on the same board. There will be a data table inside the board to store and forward the corresponding port information. By changing the port in this way, we can change the link of business data. We can make the network condition better by changing the data link and make the usage rate of network elements reach a better level, so as to better respond to the needs of the network. The other component is optical fiber. The optical fiber connects the connection between network element nodes. As a bridge between network element nodes, there may be many direct paths between two nodes

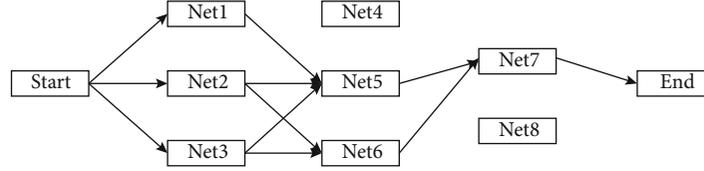


FIGURE 1: State transition expansion diagram.

to choose from. We need to follow multiple constraints. Choose the most suitable path from them to meet our network needs, thereby improving the overall network conditions. Only through the comprehensive optimization of the above two points can the overall optimization of the network topology be achieved, and our ever-growing network needs can be better. In the current development, research related to the optimization of network structure has been continuously advancing. From the structural design of the power grid to the structural design of the Internet, the realization of the bottom layer uses the research methods related to topological networks. Genetic algorithm is commonly used for the optimal routing scheme. This method can be applied to the optimization of multiple goals [2]. This method can be applied to the optimization of a single line. In the same situation, research on electric power, such as Jiang Hang's optimal structural design considering power loss and other factors, optimizes the parts in stages to improve the overall effect. For optimizing the communication delay in the network structure, the multivolume distributed subgradient optimization algorithm is used to transform the delay by expanding the dimension. Find the optimal communication situation through the adjacency matrix to achieve a better convergence effect, thereby reducing network delay [3]. It is also possible to generate a topology network algorithm through machine learning that includes a topology generation algorithm based on Lasso and its supplementary rules and the dynamic topology reconstruction algorithm by LSTM prediction [4].

The above algorithms have improved the network status to a certain extent, but in the application process, most operators still tend to use manual practical experience. This also leads to the fact that the overall network status has not been obtained. This research starts from practical applications and improves the performance of the network from three perspectives, namely, (1) starting from the situation where the network element is not restricted by the forwarding, the path planning method is proposed to the maximum; (2) individual optimization of network elements that are easily overlooked also ensures the forwarding efficiency of the network; and (3) the comprehensive consideration and the addition of manual evaluation indicators can better cope with some special scenarios.

2. Related Work

2.1. Path Planning Algorithm. The Viterbi algorithm is a very commonly used path planning algorithm. In the application process, it is mainly used in the hidden Markov model. In the model, the hidden state of the observation path is analyzed to find the maximum posterior probability of the

Viterbi path [5]. In this way, the Markov chain of the optimal solution is found in the hidden area of the joint finite state and the observable process. In the Markov hypothesis, each state behind an observed phenomenon has a probability value, and the optimal solution is found in the process; we only need to find the state with the largest probability value is the optimal solution [6, 7]. It is precisely because this state is the result probability of the superposition of the previous states; it is the state reached after multiple observations so that the fence network can be used to reflect the model well. In an ideal fence network, I use a 3×3 network structure for research. Let T be the probability of different states in a stage, so there is an observation state of $T = (t1, t2, t3)$, and the hidden state is defined as $R = (r1, r2, r3)$; through this situation, we can have the solution formula (1):

$$(r1, r2, r3) = \arg \max P(r1, r2, r3 | t1, t2, t3). \quad (1)$$

The equivalent formula (2) can be obtained by extension:

$$\arg \max \prod_{i=1}^N P(r_i | r_i) \cdot P(r_i | r_{i-1}). \quad (2)$$

The state analysis of the model is expanded as shown in Figure 1:

In the conventional calculation process, the number of combinations calculated according to this 3×3 model is 9, that is, $3N$ combinations. In the case of multiple rounds of superimposed combinations, this method requires more calculations. When using the Viterbi algorithm to solve the model, its time complexity is proportional to the length of the stage, and the complexity is $O(N * D^2)$, where N is the length of the stage and D is the width. For the application of the Viterbi algorithm, the advantages can be explained in the relevant literature: (1) If the path with the highest probability passes through a certain point of the fence network, the subpath from the starting point to this point must also be the path with the highest probability from the beginning to the point. (2) Assuming that there are k states at the i -th moment, there are k shortest paths from the beginning to the k states at time i , and the final shortest path must pass through one of them. (3) According to the above properties, when calculating the shortest path of the $i + 1$ th state, we only need to consider the shortest path from the beginning to the current k state values and the shortest path from the current state value to the $i + 1$ th state value, such as finding the shortest path when $t = 3$ is equal to finding the shortest path of all state nodes when $t = 2$ plus the shortest path of each node from $t = 2$ to $t = 3$.

In order to verify this property, we record the intermediate variable of the shortest path at the t stage as formula (3):

$$\delta_t(i) = \max P(i_t = i, i_{t-1}, \dots, i_1, o_t, \dots, o_1 | \lambda), i = 1, 2, \dots, N. \quad (3)$$

Among them, i_t is the maximum probability path at stage t , o_t is the observed value, and λ is the parameter. This formula represents the intermediate variable of a certain state, and the recursive formula (4) can be calculated by further reasoning:

$$\delta_{t+1}(i) = \max [\delta_t(j)a_{ji}] b_i(o_{t+1}). \quad (4)$$

It represents the maximum probability of the observation of o_{t+1} from the transition to state i at the $t+1$ stage, so we define o_{t+1} as the target point, which is the E node. In this process, the path nodes traversed can be expressed by formula (5):

$$\psi_t(i) = \operatorname{argmax}_{1 \leq j \leq N} [\delta_{t-1}(j)a_{ji}]. \quad (5)$$

Among them, $\psi_t(i)$ is the node through which the path with the maximum probability of state i passes when stage t . Through this formula, we can record all the nodes passed in the path of maximum probability, thereby saving the path.

The application of genetic algorithm to multitarget path search is one of the more commonly used algorithms. The main implementation process of the algorithm is divided into four steps, including initialization, selection, crossover, and mutation. In terms of implementation, given m selection targets, when applied to network path search, all possible paths need to be traversed. The time complexity is $O(N * D^2)$. According to the experiment, first choice, initial population, random choose a path with a certain node, form a chromosome set from node S to node E , select the fitness function according to different nodes, and perform genetic operations. The fitness function is based on formula (6):

$$F = \sum_{i=1}^n f_i(l_i). \quad (6)$$

Among them, $f_i(l_i)$ represents the i -th objective fitness function, and the optimal selection path of the i -th objective depends on the fitness function of l_i and the corresponding constraint information, so the value of f_i depends on l_i . In the result analysis, the lower the value of the fitness function F , the better the path. The specific experimental steps are as follows. First, for the m sets of data, we retain some excellent chromosomes and then perform cross-mutation according to the fitness functions of the respective nodes and perform next-generation operations on the new populations generated into new components and test the effectiveness of chromosomes. Finally, the solutions are sorted by the adaptability function to find the optimal path solution.

2.2. System Recommendation Algorithm. Regarding system recommendation, the recommendation system is more and more widely used at this stage, most of which are used to recommend content related to personal preferences. For users, recommendation is a specific type of intelligent information that determines the user's preferences by analyzing the user's behavioral data history and then recommends content that may be of interest to the user [8]. The main recommendation systems can be divided into two categories, content-based recommendation methods and collaborative filtering-based recommendation methods. Collaborative filtering recommendations mainly depend on users expressing their personality through ratings of movies, music, etc., which are more widely used. In this kind of recommendation, the result of recommendation is generally determined by displayed feedback and implicit feedback. The data set is formed by classifying the ratings of users and items. This type of data is displayed feedback, which is easier to collect, and its application range is also very limited [9]. Another kind of feedback is achieved through the user's preferences for items. This type of feedback is implicit feedback. In this process, implicit feedback is only negative and positive, so analysis is more difficult. In general, for implicit feedback, we use a pairwise ranking method for ranking. In the pairwise comparison, we can know the preference order relationship between different items. In the application of pairwise ranking, the optimal ranking algorithm is a method of generating personalized recommendations based on implicit feedback data, which is Bayesian personalized recommendation (BPR) [10]. In this recommendation method, it mainly treats all unobserved feedback as negative feedback, treats all observed items as positive examples, and assumes that the user's preferences are independent; it assumes that users prefer positive feedback items, not items without positive feedback. BPR groups the data according to the association and then ranks the group's data into positive or negative effects. Its purpose is to maximize the distance between observed and unobserved interactions. Formally, the objective function of BPR is formula (7):

$$L_{\text{BPR}} = \sum_{(u,i,j) \in D} -\ln \sigma(\hat{x}_{u,i} - \hat{x}_{u,j}) + \lambda \|\Theta\|^2. \quad (7)$$

Among them, $\hat{x}_{u,i}$ represents u 's preference for item i in a set of data, $\sigma(x)$ is a sigmoid function, λ is a model-specific regularization parameter to prevent overfitting, and Θ is a collection of multiple training pairs in the formula. Thus, the recommendation list result is obtained according to $\hat{x}_{u,i}$. However, in the rating or ranking, the projects in which users do not participate account for the vast majority of projects. We can find that it is flawed in the application process. In order to make this research more effective in optimizing the network, we introduce a Mean Bayesian Personalized Ranking Algorithm (MBPR), which converts the displayed feedback data into implicit feedback data.

For the data processing of the MBPR method, the method defines the observed interactions above the average score as positive feedback and marks it as +1, and the observed

interactions below the average score as negative feedback and marks it as 0, mark the unobserved interaction as a random value between (0,1). We give a parameter matrix formula about W and H according to Bayesian posterior and Bayesian formula. W and H decompose all item rating matrices into preference matrix W and item rating matrix H .

$$p(\Theta | >_u) \propto p(>_u | \Theta)p(\Theta). \quad (8)$$

Among them, $>_u$ represents the total sorting scheme of the items selected by the user. We assume that the item selection scheme of each project is completely independent, and we can get formula (9):

$$\prod_{u \in U} p(>_u | \Theta) = \prod_{(u,i,j) \in D_s} p(i >_u j | \Theta). \quad (9)$$

Among them, i and j , respectively, represent the two option values that are compared pair by pair in the item selection, that is, the comparison result of i and j when simulating the user's selection of the item. From this, we can see that the order of j is higher than the order of i (10):

$$p(i >_u j | \Theta) = \sigma(\hat{x}_{u,i,j}(\Theta)). \quad (10)$$

Among them,

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (11)$$

$\sigma(x)$ is the sigmoid function. We use this formula as the loss function, and in this way, the item selection of different network elements is mapped to hidden factors, and recommendations are formed through these factors. In this process, we also need to determine the maximum posterior probability of the parameters W and H . In citing the MBPR method, its objective function formula (12):

$$L_{\text{MBPR}} = \ln p(\Theta | >_u) \propto \ln p(>_u | \Theta)p(\Theta). \quad (12)$$

As a result, we began to use the method to apply the data and train the model to determine the W and H parameters. Use the SGD method in the literature to learn parameters, according to formula (13):

$$\begin{aligned} \frac{\partial L_{\text{MBPR}}}{\partial \Theta} &= \sum_{(u,i,j) \in D_s} \frac{\partial}{\partial \Theta} \ln \sigma(\hat{x}_{u,i,j}) \\ &+ \lambda \frac{\partial}{\partial \Theta} \|\Theta\|^2 \propto \sum_{(u,i,j) \in D_s} \frac{-e^{-\hat{x}_{u,i,j}}}{1 + e^{\hat{x}_{u,i,j}}} \times \frac{\partial}{\partial \Theta} \hat{x}_{u,i,j} + \lambda \Theta, \end{aligned} \quad (13)$$

where

$$\frac{\partial}{\partial \Theta} \hat{x}_{u,i,j} = \begin{cases} h_{i,f} - h_{j,f} & \text{if } \Theta = w_{u,f}, \\ w_{u,f} & \text{if } \Theta = h_{i,f}, \\ -w_{u,f} & \text{if } \Theta = h_{j,f}, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

After constant iteration, the W and H parameters become more accurate, and the fit is better. After the parameters are determined, we have established a complete MBPR recommendation model.

3. Application

3.1. Path Planning Realization. This research is based on improving the overall structure of the network topology to meet the growing demand. The specific implementation logic diagram is shown in Figure 2.

In this method, the optimization of the overall network structure is mainly achieved by optimizing routes and network element nodes, and the optimization of the network is evaluated through evaluation tools. Applying the Viterbi algorithm to the optimization of the network topology, we select the network conditions of a certain city for experiments and perform local optimization for the lines that reach the critical value.

In terms of path optimization, the network conditions are optimized mainly from the following starting points:

- (1) The main and standby routing rates are the same. In order to ensure a more stable service path, the main and backup paths of the service path are optimized to make them more highly available
- (2) The huge ring node ratio ensures that the service path is a shorter path and avoids the increase in data transmission delay
- (3) Link CIR bandwidth occupancy rate, the occupied bandwidth of a single line traffic should not be too large
- (4) Through the optimization of the above four indicators, from the network element service carrying rate, due to the limited number of services carried by a single network element node, in order to ensure the safety of the network element node, make it in a relatively suitable utilization rate range

Through the optimization of the above four indicators, the system structure of the entire topology network is improved, so as to better meet the needs of future network development for the network structure. The demand analysis shows that the research has important significance and practical value.

When planning the path, use the model of the simulated fence network to establish a similar network connection diagram. The network element node is the junction point of the fence network, and the connection between the network

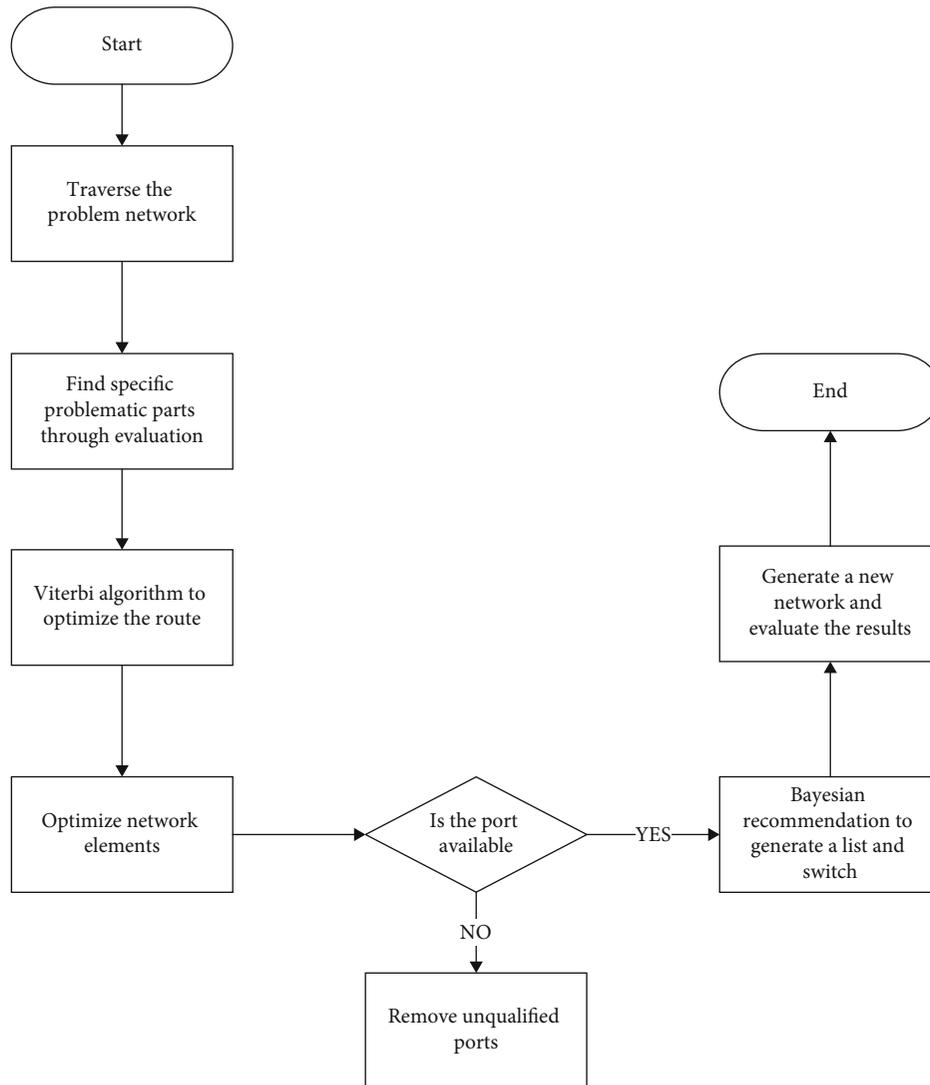


FIGURE 2: Implementation plan.

element and the network element is routed according to the connection of the optical fiber.

The second thing that needs to be considered is the evaluation tool based on network conditions. The main considerations for a single line are the problem of active and standby routes, network element service carrying rate, giant ring node ratio, and CIR broadband occupancy rate. For all reference factors and other issues, the route is dynamically scored in the form of weighting and penalty [11]. For the problem of active and standby routes, if the primary path and the backup path overlap, the network element node will be punished, and the rated traffic of the network element should not be exceeded. No processing is performed under 80% of the cases, and the load exceeding 80% of the rated capacity will be punished. Juhuan's judgment is that the number of network elements on a single line is calculated. Therefore, with the number of network elements node on a single line calculated, the more network elements node, the lower the score. The CIR broadband occupancy rate mainly considers the service data carried by a single fiber. The higher the occupancy

rate, the lower the score. Establish a path evaluation model through this form and then select the optimal path through the Viterbi algorithm.

Based on the application of the Viterbi algorithm in the network research system, the nodes starting from a single convergent network element are not balanced in the same hidden state at the same stage in the path process, and some special nodes cannot be used too much. For optical fibers and some other unusable network elements, we adopt a negative feedback mode to make the selected path more reasonable. Through the above weighted parameters and related algorithms, we have established a path planning model.

In the optimization applied to the topological network structure, firstly, the network data is preprocessed, and the optical fibers of the entire network structure are first traversed to establish a complete regional topological network diagram. As the area may be large, we select some parts for display, as shown in Figure 3:

In a certain area, the approximate network element distribution is shown in the above figure. According to the

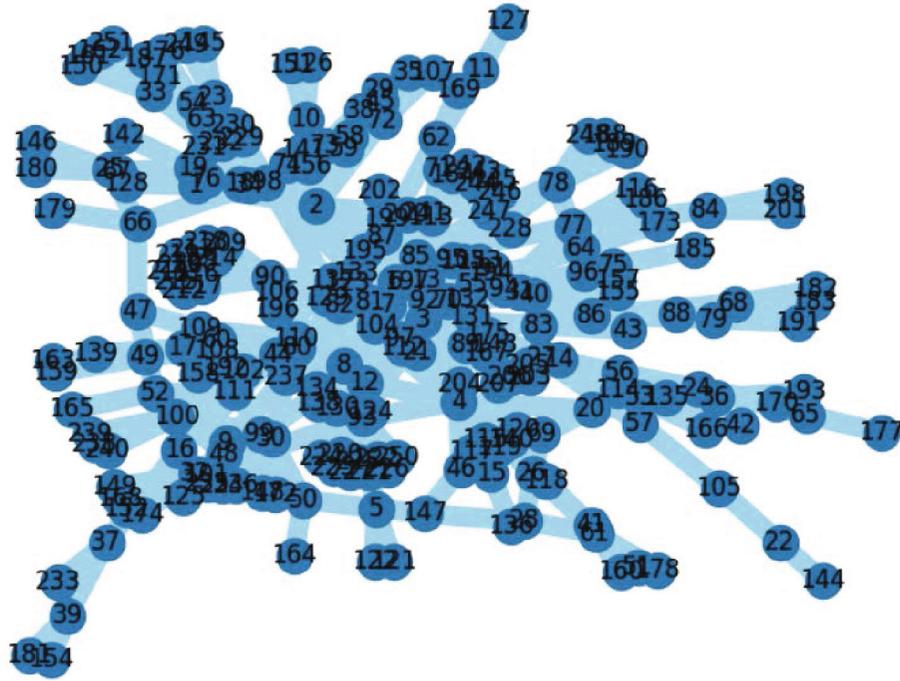


FIGURE 3: Partial visualization of topological network.

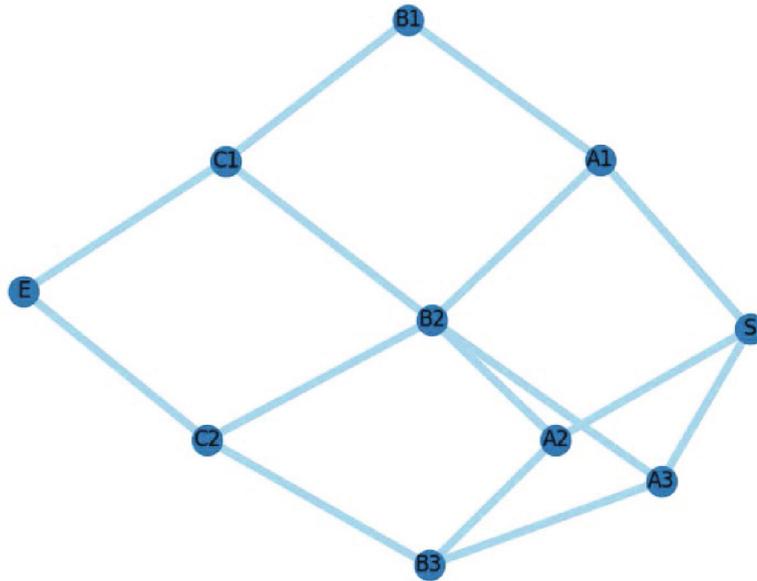


FIGURE 4: Typical structure diagram.

evaluation report of the entire network, we find out the network elements that need to be improved. We select the more typical cases to analyze and the application of the model. In this study, we mainly optimize the weak parts of the entire topology network. There may be many different situations, but because in the topology network architecture, we mainly optimize the connection between the convergent network elements, so we choose a more typical network structure as an example for experimental verification.

The actual circuit diagram model related to the construction of optical fiber is shown in Figure 4:

In this scenario, the two network elements S and E are the starting and ending network element nodes, and the other network element nodes are nodes on the path. Since only a part of the entire topological network is selected for analysis, the network needs to be initialized. Each node in the topological network may exist in other paths. In order not to damage other local network conditions, we need to the initial status of

a network element is initialized and weighted, and the score is weighted according to the current business volume and CIR occupancy of the network element. For special dedicated line network element nodes, they are removed.

In the application process, the main path is mainly maintained, and the backup path is an optional item. Therefore, the backup path that is not interfered by the main path is selected under the condition that the main path is optimal. In the course of the experiment, we need to consider the three characteristics of network element service carrying rate, giant ring node ratio, and CIR broadband occupancy rate to select the optimal path. In order to ensure the reliability of the experiment, genetic algorithm is introduced for experiment comparison [12]. The excellent primary path is selected, and the backup path is sorted in the same way to select the better path, thereby determining the primary path and the backup path.

For using the Viterbi algorithm, find the optimal path as the main path [13–17]. In the process of finding the main path, that is, starting from the network element of the sink node S , find the optimal path to the network element of the sink node E and find an optimal solution for the main and backup paths. Before optimizing, we start from node S , and there are multiple nodes in the node connecting S . We use the number of nodes to reach S node as the intermediate state. According to the formula, the node extending from node S will reach one at each stage; there will be a corresponding optimal solution with the maximum probability, and the optimal path from S to E is found by the continuous recursion of the model.

After the optimal path is generated, the corresponding path will be temporarily recorded, and the backup path will be searched according to the record, and the network element will be initialized again based on the main path search. In order to avoid the same network element line as much as possible, for the selected network element, the value is passed into the model in the form of negative feedback, and then, the alternate path selection is performed, and the optimal path selection is performed in a manner similar to the main path selection. When the temporarily recorded path appears, the corresponding penalty, thereby reducing the score, has reached the final priority path.

After achieving path optimization, we also need to plan specific network element nodes to ensure the normal and efficient operation of the nodes to improve overall performance. In this process, the most important thing is the selection of the access and output ports of the network element. After judgment, this is more suitable for the recommended algorithm.

3.2. Network Element Optimization. In this research, network elements undergo a reasonable distribution of transaction for a balanced traffic to prevent single node downtime. Two Bayes recommendation algorithms are introduced for experimental comparison, and intelligent decision-making is performed to allocate transaction. In practice, feedback data in Bayes recommendation algorithm can be mainly divided into three categories: At the same time, two Bayesian recommendation algorithms are introduced for experimental comparison,

and intelligent decisions are made to allocate some business conversions. In the application process, in the Bayesian recommendation algorithm, the feedback data can be mainly divided into three categories: (1) the rating of the local network by the network status scoring tool, (2) resource usage in specific network elements, and (3) the topological network engineer's rating of the port allocation and operation of a single network element.

For using BPR, the unobservable data is negative feedback, and the observable data is positive feedback. In this case, the second type is explicit data, which is positive feedback, and the first and third types are implicit data, which are understood as negative feedback [17, 18]. Because the three types of data are more closely related, they are more suitable for data feedback using the converted MBPR method. The data processing of this method is well adapted to this application [19]. In this application, it is mainly to recommend an ideal port selection plan from a network element. We can assume that the selected port is an item. The user makes a decision on the port selection through three types of data feedback. Therefore, our training data screened based on the best performing data among the three types of feedback and conducted training with manual selection of the results. After screening, we obtained the selection of ports and boards among 1,000 network elements. The status of the network elements as well as the information related to the boards and ports is graded and put into the training model, respectively.

After establishing the model, we began to conduct application experiments. According to the application analysis of using Bayesian recommendations in network element optimization, we used the graph process to conduct experiments, as shown in Figure 5.

First, we grab the relevant data of the network element from the running network of a certain city and then preprocess the data and collect the relevant data according to the way of dividing the data into three categories. For the first type of data, we use a general network condition assessment tool. Because it is an overall scoring system, only parts can be rated. Then, we need to rate a single network element node, so we assume that the local rating represents the average rating of all network elements in the area. To ensure the integrity of the data, we set the local area rating as the rating of all the network elements in the area for evaluation. For the second type of data, since there are many indicators related to the status of the network element, it mainly includes port resource utilization, the same board rate of the active and standby network elements, traffic balance, link bandwidth occupancy rate, network element service load rate, network metarate, etc.; for the above indicators, we use the form of clustering to reduce the dimensionality of multiple indicator data. In the process of dimensionality reduction, it is found that three of the indicators account for a relatively high proportion. These are also important indicators that we need to solve.

- (1) *Port Resource Utilization.* For the current network element, there are multiple ports, and the port selection of the service should be allocated reasonably to make the network more secure

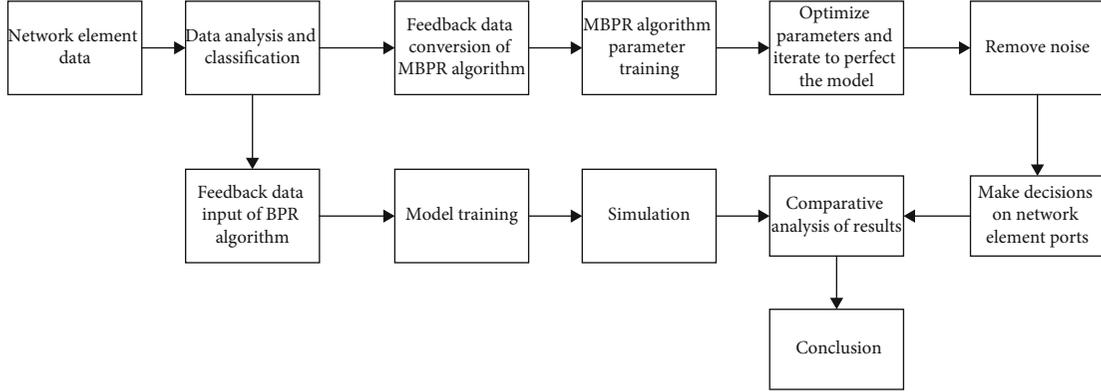


FIGURE 5: System recommendation algorithm flow design.

- (2) The main and backup network elements are on the same board. In the case that the main and backup paths cannot be the same path and when the two paths pass through the same network element at the same time, the connection cannot appear on the same board. Avoid data confusion and instability due to accidents
- (3) *Flow Balance Degree*. For each board in the network element, it is necessary to achieve a balanced flow state as much as possible. If the local flow is too large, it will also cause the failure of the network element node

Therefore, we analyze the need to pay attention to three indicators in the network element.

In practical applications, ports in the same board can forward related services within the same board according to the built-in data table. We parameterize the preprocessed data to form the second type of data parameters. For the third type of data, he is mainly based on the evaluation of the relevant network elements by engineers and experts. These evaluations involve the performance of the network elements in the actual application process. In this evaluation system, the network elements are mainly divided into two. The categories are qualified and to be improved. Through the comprehensive statistics of the three types of data, we have formed three types of feedback information on the source of the network element evaluation information. Then, analyze the feedback information according to Bayesian' basic formula (15):

$$P(B_i | A) = \frac{P(B_i)P(A | B_i)}{\sum_{j=1}^n P(B_j)P(A | B_j)}. \quad (15)$$

For the application of the BPR algorithm, the data is grouped based on maximizing the separation of observed feedback and unobserved feedback, the second type of data is added to the model training as positive feedback, and the first and third types of data are added to the model as negative feedback training.

Thanks to the independence of preference behaviors between items, the partial order of different ports of the same item is independent of each other. We treat each piece of data

as an independent part to improve the effectiveness of the feedback data. Put the processed sample data into the model for training and follow the code design drawing for training.

According to the objective function of BPR, after gradient descent, we calculate the corresponding loss function and adjust the corresponding weights of positive feedback and negative feedback in the iterative process, and when the loss function is minimized, we input test data and get the corresponding result.

For the application of the MBPR algorithm, according to the maximum posterior of the Bayesian algorithm, it is mainly to determine the matrix of W and H parameters. We map the second type of data to the factors of the port by means of mapping, and then according to the posterior, analyze the first and third types of data. Through the training of sample data, iterate continuously according to the formula. In the iterative process, the parameters are initialized many times to avoid the phenomenon of parameter overfitting. We introduce parameter iteration formula (16); after constant initialization of parameters, the parameters are better.

$$\Theta \leftarrow \Theta + \eta \left(\frac{-e^{\hat{x}_{u,i,j}}}{1 + e^{\hat{x}_{u,i,j}}} \times \frac{\partial}{\partial \Theta} \hat{x}_{u,i,j} - \lambda \times \Theta \right). \quad (16)$$

And to avoid some disturbance factors, in this process, the noise is continuously filtered out. The main source and port value are partially exceeded. When the $\sigma(x)$ loss function is the smallest, we obtain the corresponding parameters and then use the test data for correlation experiment of.

4. Analysis of Experimental Results

4.1. Path Planning Experiment Analysis. In order to verify the experimental results, we select a local network for route planning experiments, use genetic algorithms for experimental comparison, analyze the calculation time, and compare the coincidence rate of the path. The result is shown in Table 1:

Through the comparison of the two experimental methods, we found that the sample data shows that the coincidence node rate of the primary and backup paths using the genetic algorithm is significantly higher than that of the improved Viterbi algorithm. In order to ensure the stability

```

Input Training samples, number of iterations n, regularization parameters
Output model parameters
  Initialization parameters
  While i<n do
    according to  $P(B_i | A) = P(B_i)P(A | B_i) / \sum_{j=1}^n P(B_j)P(A | B_j)$  Maximum a posteriori  $P(W,H|>u)P(W,H|>u)$  to
    solve the model parameters W, H, use the sigmoid function to set the threshold
  End
Return Recommended list

```

ALGORITHM 1: BPR designing process.

TABLE 1: Algorithm performance comparison.

Algorithm	Calculation time	AVG network element overlap
GA	0.251	5
Viterbi	0.011	2

TABLE 2: Comparison of experimental data generated by BPR and MBPR.

Method	Data	Port	Sparsity	Levels
BPR	1000	26241	85.40%	3-5
MBPR	1000	26241	92.66%	4-5

of the network and avoid the network failure due to a single line problem, therefore, it is better to use a scheme with a low number of network elements, so the Viterbi algorithm is better. In terms of calculation time, the Viterbi algorithm effectively reduces the time complexity due to its core idea of dynamic programming. Therefore, the Viterbi algorithm performs better in large-scale application scenarios. Comprehensive comparison Viterbi is more suitable for network path planning than genetic algorithm.

4.2. Network Element Optimization Experiment Analysis. In order to ensure the effectiveness of network element optimization, we select 1000 sets of data for experiments to compare the efficiency of the Bayesian recommendation algorithm. After comparing the experimental data, see Table 2.

Through the analysis of sample data, the results produced by the two schemes can be used for reference. In terms of the posterior nature of the joint evaluation tool scoring and the engineer's rating, the port score recommended by the MBPR method is better. During the experiment, MBPR has added some parameters to prevent data disturbance and has undergone denoising processing, so the time complexity is lower than that of BPR. When using the BPR method for recommendation, the method is mainly to obtain the recommendation list result according to $\tilde{x}_-(u, i)$. When the user does not participate in the project, the effect will not be ideal. In this application, more attention is paid to the first category compared with the posterior data of the third type of data; the weight of the second type of data is low, and it is cited in negative feedback, so there are some unsatisfactory conditions.

And when its effectiveness is judged by the AUC index, its comparative effect is shown in Figure 6.

Through the analysis of the sorting accuracy in the sample data, we can make a selection according to different scenarios. In large-scale application scenarios and when computing resources are lacking, we can use the BPR method to improve efficiency. In order to pursue better performance, it is more recommended to use the MBPR method for recommendation ranking.

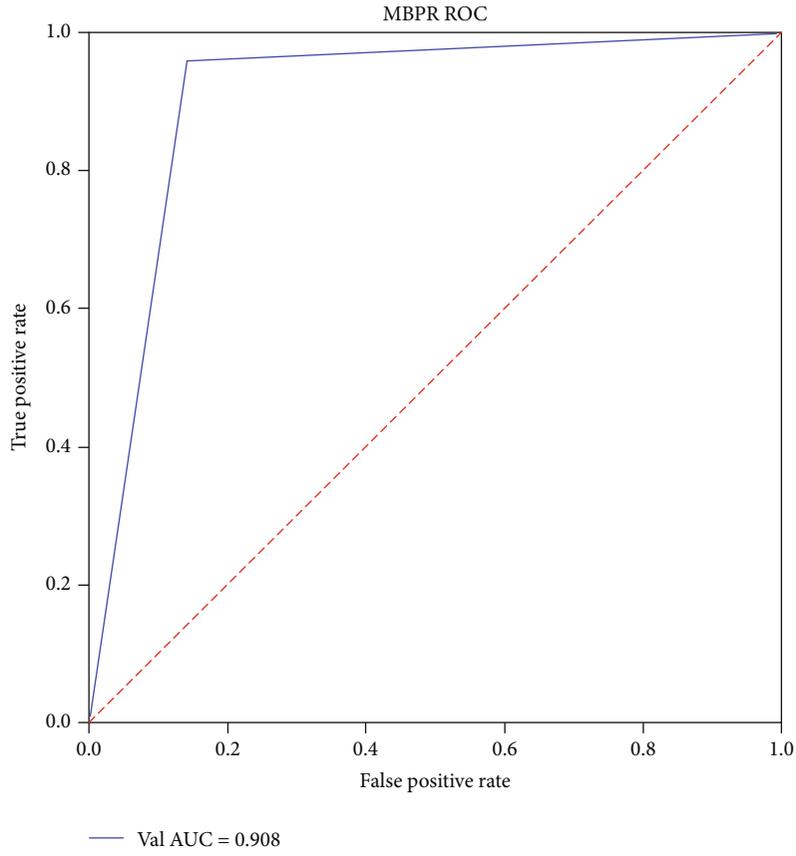
In evaluating the effect of the recommendation algorithm, we introduce a model evaluation program. When optimizing the port, it is important to establish a reliable recommendation list by selecting and recommending the board and port forwarded by the network. For the model training of the selected 1000 sets of data, each network element will have a corresponding initial value of the network element condition. Since the initial value of different network elements may have large differences, it is likely that serious defects will occur when the model is overfitted. Therefore, when collecting data, choose as large a difference as possible sample and guarantee the difference of the sample. In order to verify the effectiveness of the recommendation algorithm, we use a general model evaluation method to verify the port ranking through accuracy. This evaluation method has been widely used in the system to recommend evaluation indicators. Specific data refer to formula (17):

$$\text{Precision} = \frac{1}{\text{net}} \sum_{u=1}^{\text{net}} \frac{\text{utilization} * 0.4 - \text{samerout} * 0.4 + \text{flow} * 0.2}{\text{origin}}. \quad (17)$$

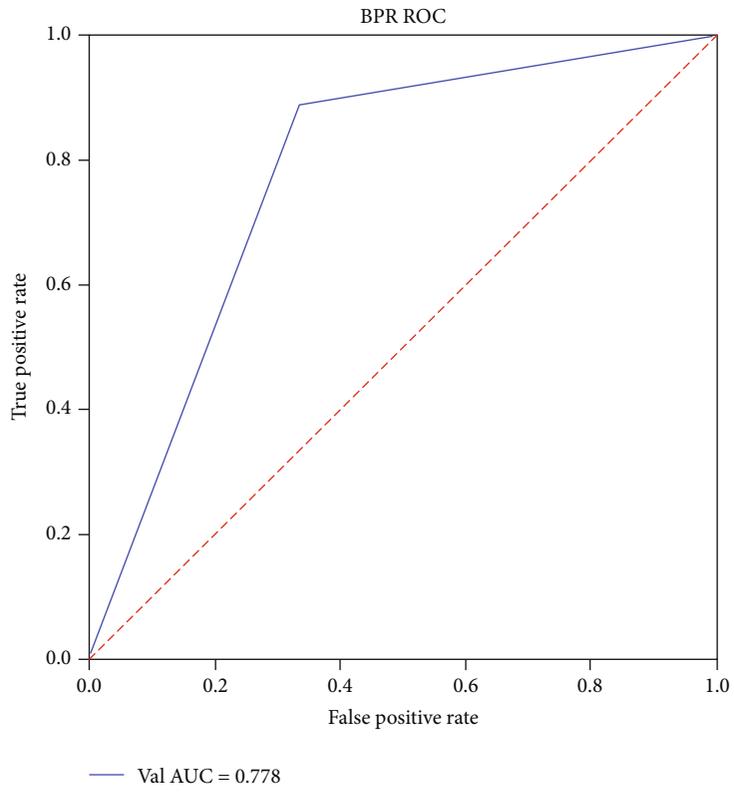
Among them, port_{new} represents the recommended port score, $\text{port}_{\text{origin}}$ represents the original port score, and origin represents the original data. The accuracy of the original data and the optimized data are compared to evaluate the effect of the model. The corresponding results of the two algorithms were verified by the prior data indicators, and the results are shown in Figure 7.

Through the change of the relevant parameter value, we get the change of accuracy as shown in Table 3.

During the experiment, in order to ensure that the data can reach the convergence effect, we uniformly set the number of iterations to 10000, and in the same 1000 samples, set 300 sets of data as training samples and 700 sets as test samples. After testing, the recommendation accuracy rate based on the MBPR method is higher, and then, different types of



(a) MBPR experiment



(b) BPR experiment

FIGURE 6: AUC curve.

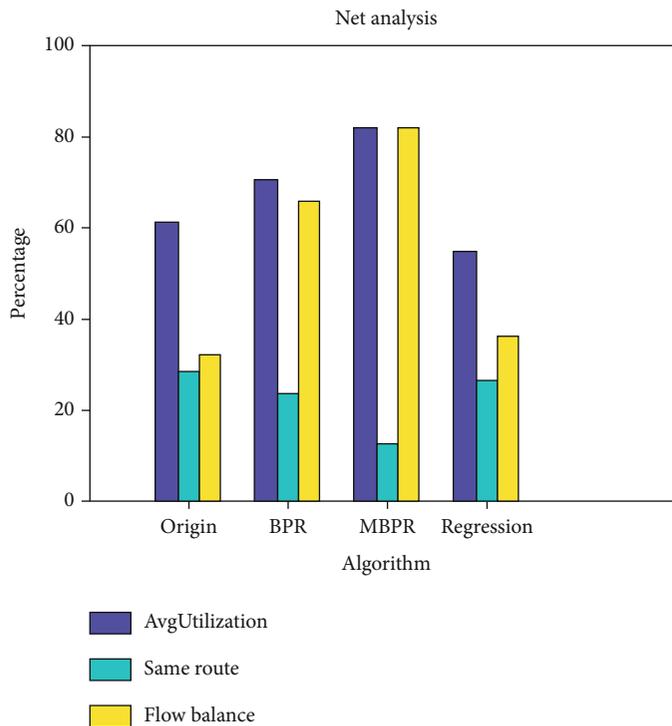


FIGURE 7: Comparison of changes in the three main indicators of optimization.

TABLE 3: Model evaluation parameters.

Algorithm	Precision
BPR	76.20%
MBPR	85.03%
Regression	62.66%

feedback data are analyzed, and the prior data is compared with the posterior data. In the baseline comparison result of the overall data, the prior data weight is higher. In the posterior feedback data, the impact coefficient of the data set based on the evaluation tool is higher than that of the manually rated data. The result obtained by using the regression model and the prediction method is very small, and there is a decline in some indicators. After studying the literature related to this article, add relevant theoretical and experimental analysis [20–24]. Comprehensive consideration, the use of MBPR performs better.

5. Conclusion

In this paper, we propose a near optimal routing scheme between two network elements by different weighting forms and dual Viterbi algorithm. Recommended value was selected, and optimization results were obtained by average Bayes algorithm. A set of solutions that meet requirement of efficient and balanced network elements were achieved. It is shown that the proposed routing scheme achieves a significant performance improvement in despite of accuracy of the algorithm that still needs to be improved. The proposed

scheme can be implemented with low computational cost and is efficient for large-scale topology networks.

Data Availability

The source of the basic data and the network data of a certain city is inconvenient to disclose due to confidentiality.

Conflicts of Interest

The author declares that he/she has no conflicts of interest.

Acknowledgments

This work is funded by the National Natural Science Foundation of China under Grant No. 61772180.

References

- [1] L. I. Guo, Z. U. O. Xue-qi, and W. E. I. Yu, “The selection strategies of 5G NSA anchor,” *Digital Technology & Application*, vol. 38, no. 12, pp. 26–28, 2020.
- [2] X. C. TianLu and L. J. SunJianwei, “Topology optimization design of time TDMA inter-satellite link based on heuristic genetic algorithm,” *Computer Measurement & Control*, vol. 28, no. 12, pp. 155–160+171, 2020.
- [3] L.-Q. Liujun, “Distributed subgradient method for multi-agent optimization with communication delays,” *Journal of Hefei University of Technology*, vol. 36, no. 5, pp. 559–565, 2013.
- [4] L. Xiaoyu, *Research on Topology Generation and Reconfiguration Optimization of Distribution Network Based on Machine Learning*, Beijing University of Posts and Telecommunications, 2019.

- [5] A. Caliebe, "Properties of the maximum a posteriori path estimator in hidden Markov models," *IEEE transactions on information theory*, vol. 52, no. 1, pp. 41–51, 2006.
- [6] O. Cappé, E. Moulines, and T. Rydén, *Inference in hidden Markov models*, Springer, 2005.
- [7] S. Derrode and W. Pieczynski, "Unsupervised data classification using pairwise Markov chains with automatic copulas selection," *Computational Statistics & Data Analysis*, vol. 63, pp. 81–98, 2013.
- [8] I. T. Christou, E. Amolochitis, and Z.-H. Tan, "AMORE: design and implementation of a commercial-strength parallel hybrid movie recommendation engine," *Knowledge and Information Systems*, vol. 47, no. 3, pp. 671–696, 2015.
- [9] W. Zhou, J. Li, Y. Zhou, and M. H. Memon, "Bayesian pairwise learning to rank via one-class collaborative filtering," *Neurocomputing*, vol. 367, pp. 176–187, 2019.
- [10] J. Ding, G. Yu, X. He, F. Feng, Y. Li, and D. Jin, "Sampler design for bayesian personalized ranking by leveraging view data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 667–681, 2021.
- [11] J. Jo, H.-G. Kim, I.-C. Park, B. C. Jung, and H. Yoo, "Modified viterbi scoring for HMM-based speech recognition," *Intelligent Automation & Soft Computing*, vol. 25, no. 2, pp. 351–358, 2019.
- [12] H. B. Su, Y. L. Shi, and Z. Z. Hou, "Multiobjective and multipath optimization selection methods based on genetic algorithms," *Microelectronics & Computer*, vol. 23, no. 10, pp. 41–43, 2006.
- [13] J. Lember and J. Sova, "Existence of infinite Viterbi path for pairwise Markov models," *Stochastic Processes and their Applications*, vol. 130, no. 3, pp. 1388–1425, 2020.
- [14] J. Lember, K. Kuljus, and A. Koloydenko, "Theory of segmentation," in *Hidden Markov Models, Theory and Applications*, P. Dymarski, Ed., pp. 51–84, InTech, 2011.
- [15] J. Hamilton, *Regime Switching Models, in: Macroeconometrics and Time Series Analysis*, Springer, 2010.
- [16] K. Kuljus and J. Lember, "On the accuracy of the MAP inference in HMMs," *Methodology and Computing in Applied Probability*, vol. 18, no. 3, pp. 597–627, 2015.
- [17] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski, "Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise," in *Signal Processing*, vol. 91, no. 2pp. 163–175, Elsevier, 2011.
- [18] W. Pan, H. Zhong, C. Xu, and Z. Ming, "Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks," *Knowledge-Based Systems*, vol. 73, pp. 173–180, 2015.
- [19] J. F. Wang and P. Han, "Adversarial training-based mean Bayesian personalized ranking for recommender system," *IEEE ACCESS*, vol. 8, pp. 7958–7968, 2020.
- [20] Y. Zhai, D. Liu, C. Wu, and R. She, "A recommendation approach based on Bayesian networks for clone refactor," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1999–2012, 2020.
- [21] C. Cai, H. Xu, J. Wan, B. Zhou, and X. Xie, "An attention-based friend recommendation model in social network," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 2475–2488, 2020.
- [22] S. Bin, G. Sun, N. Cao et al., "Collaborative filtering recommendation algorithm based on multi-relationship social network," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 659–674, 2019.
- [23] W. Jiang, J. Chen, Y. Jiang et al., "A new time-aware collaborative filtering intelligent recommendation system," *Computers, Materials & Continua*, vol. 61, no. 2, pp. 849–859, 2019.
- [24] Y. Shen, Y. Li, J. Sun et al., "Hashtag recommendation using LSTM networks with self-attention," *Computers, Materials & Continua*, vol. 61, no. 3, pp. 1261–1269, 2019.