Wireless Communications and Mobile Computing

Intelligent Edge Computing for Future Communications

Lead Guest Editor: Peiying Zhang
Guest Editors: Chunxiao Jiang and Maozhen Li

Intelligent Edge Computing for Future Communications

Wireless Communications and Mobile Computing

Intelligent Edge Computing for Future Communications

Lead Guest Editor: Peiying Zhang
Guest Editors: Chunxiao Jiang and Maozhen Li

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9816845, 1 page
https://doi.org/10.1155/2023/9816845

Retraction

Retracted: Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning

Wireless Communications and Mobile Computing

Received 8 August 2023; Accepted 8 August 2023; Published 9 August 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1].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) Peer-review manipulation

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] M. Li, W. Sun, Q. Jia et al., "Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1931965, 7 pages, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 2753085, 13 pages
https://doi.org/10.1155/2023/2753085

Research Article

A Network Fault Prediction-Based Service Migration Approach for Unstable Mobile Edge Environment

Haiyan Wang, Weihao Tang, and Jian Luo

Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China

Correspondence should be addressed to Haiyan Wang; wanghy@njupt.edu.cn

Received 26 September 2022; Revised 11 November 2022; Accepted 3 June 2023; Published 8 August 2023

Academic Editor: Peiying Zhang

How to perform efficient service migration in a mobile edge environment has become one of the research hotspots in the field of service computing. Most service migration approaches assume that the mobile edge network on which the migration depends is stable. However, in practice, these networks often fluctuate greatly due to the fault of edge devices, resulting in unexpected service interruptions during the migration process. Besides, most of the existing solutions do not consider the migration cost and path selection in the event of edge network fault. Aiming at the above problems, we propose a service migration approach based on network fault prediction (SMNFP) for mobile edge environment. The SMNFP method first introduces the software-defined network as a global controller, which is used to monitor and collect the changing of the edge network and schedule the migration tasks. Second, a network fault prediction model based on Wide&Deep model is proposed to predict the upcoming faults in the network according to the alarm information of network equipment. Finally, the service migration problem is constructed as a Markov decision process, and a fault penalty function is introduced to avoid faulty nodes, together with the deep *Q*-learning method to solve the migration strategy. Simulation experiments are conducted on the public metro network fault dataset, and results show the proposed method can effectively predict network faults and complete service migration.

1. Introduction

In recent years, the development of mobile Internet has enhanced the performance of the network, such as bandwidth, transmission rate, and throughput rate. Mobile edge computing (MEC) allows us to deploy servers geographically closer to users, provide computing power closer to smartphones or various types of mobile terminals, and sink these computing power into base stations. However, in MEC, the limitations of edge server coverage, the mobility of edge end users, and the differences in mobile requests in different regions often cause load imbalance between servers, which in turn leads to service quality degradation and even service interruptions [1, 2].

To ensure the continuity of services when users move, service migration technology has begun to receive extensive attention. In the mobile edge network scenario, service migration refers to migrating the application services used by the user from the connected edge server according to a certain algorithm or decision-making mechanism under the premise of ensuring the minimum cost and delay during the rapid movement of the user to the best server at different times.

Existing service migration methods usually assume that the user's moving path is known [3, 4], and some research work has predicted the user's mobility, using mobility prediction and perception methods to carry out their work [5, 6], in which services are premigrated to relevant areas, effectively reducing the migration workload. In addition, some researchers use Markov decision process to model service migration [7, 8], reducing the overall computational complexity of the model.

However, we emphasize that there are two main problems with the previous methods. One is that these methods assume the edge network in which they migrate is in a stable condition, and these methods are valid with a limitation to the fault-free edge network. In reality, the edge network is unstable. As shown in Figure 1, edge devices may cause the

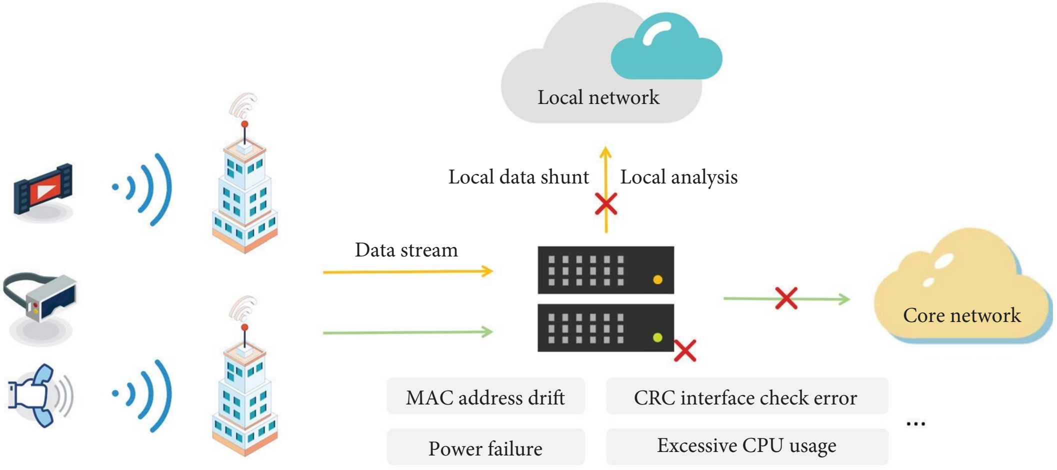


Figure 1: A network fault scenario caused by equipment faults.

temporary failure of edge nodes due to various reasons. The other is that most of the previous studies only consider the migration path selection problem in intact edge networks, lacking the collection and aggregation of all migration tasks and network topology information in edge networks and thus lacking a unified real-time scheduling means for service migration for different network situations.

To address the above issues, we employ the software-defined network (SDN) mechanism to predict the failures of the edge network. We then put forward the service migration approach based on network fault prediction (SMNFP) method to circumvent the faulty nodes and make reasonable migration path selection. The main contributions of this paper are as follows:

(i) First, we proposed an edge network fault prediction module network fault prediction model based on Wide&Deep model (NFP-WD), which is used to predict the fault of the entire edge network within a fixed time window and mark all edge servers that may fail before the next time window.

(ii) Second, we present the SMNFP method, in which the service migration problem is constructed as a Markov decision process, and a fault penalty function is designed to avoid faulty nodes in the migration path selection. Finally, the deep Q-learning method is used to solve the service migration strategy.

(iii) Finally, we introduced the SDN framework as the migration controller for the model as a whole. Then we conduct simulation experiments on the MAN fault data set, and the experimental results show that our SMNFP method has a better migration effect than the baseline method.

2. Related Work

2.1. Service Migration. In terms of service migration, many researches and methods are aimed at cost balancing and optimization in the migration process. Liang et al. [9] used a combinatorial optimization algorithm and integer relaxation iterative algorithm to optimize the offload rate, mobility, and MEC throughput of services in cellular networks, which indirectly reduced the migration cost. Wang et al. [10] investigate a user-centric service migration and exit point selection problem which introduces a neural network-based smart migration judgment to navigate the performance and computation overhead tradeoff. Park et al. [4] formalized the migration cost, communication cost, and energy consumption associated with the migration process as a complex optimization problem, employing deep reinforcement learning to approximate the optimal policy. Wang et al. [8] designed a service migration framework Mig-RL by using the reinforcement learning method; when encountering similar migration patterns, the migration strategy can be directly retrieved, which significantly reduces the decision-making cost.

Another part of the research work is based on user mobility perception and prediction. Yin et al. [5] proposed a mobility-aware service migration mechanism, which selects the target node for migration according to the migration cost and the moving direction. Labriji et al. [6] used the mobility prediction method to solve the problem of vehicle service migration. The method combines neural network and Markov chain for vehicle mobility prediction, which can still maintain a good performance in the scene of large-scale traffic flow. Xu et al. [11] proposed a service migration method based on the Bernoulli test and made a quantitative analysis of delay and user mobility prediction through theoretical analysis and simple probability statistics, which effectively reduced service communication delay and migration cost. Miao et al. [12] proposed a mobility-enabled service migration scheme, called MSM, for real-time decision-making on service migration.

2.2. Network Fault Prediction. At present, the related research on network fault prediction is mainly based on the methods of machine learning and deep learning. In terms of machine learning, Lin et al. [13] predicted faults in smart distribution networks by introducing multiple support vector machines (SVMs) and an improved voting random forest algorithm, which improved the accuracy rate and recall rate. Yadwad and Vatsavayi [14] combine hidden Markov models with Bayesian networks for outage prediction of network devices. In terms of deep learning, Google's Wide&Deep model [15] is widely used in the field of recommender systems; some researchers [16] used the Wide&Deep model to carry out network fault prediction work. In addition, Klein et al. [17] used two-dimensional convolutional neural networks to extract temporal feature data streams and then used graph convolutional neural networks to extract spatial features, combined with domain expert knowledge to jointly predict network faults. Tefera et al. [18] used long short-term memory network (LSTM) and gated recurrent unit for early prediction of base transceiver stationfaults caused by power system and environmental anomalies.

2.3. Deep Reinforcement Learning. Reinforcement learning is an important machine learning method. It takes the next action based on the feedback of the environment, through constant interaction and trial and error with the environment, and achieves the final goal in the case of obtaining the maximum benefit as a whole. Q-learning is a typical reinforcement learning method based on Q value. Wang et al. [8] leveraged the Q-learning approach to design a service migration framework for reducing the total service cost in mobile edge environments. However, in the actual service migration environment, the edge network environment is relatively complex, which easily leads to large state space. Therefore, it is unrealistic for reinforcement learning to store action values through a Q table. To solve this problem, Mnih et al. [19] first combined the convolutional neural network and Q-learning method and proposed a deep Q-network (DQN) model for processing visual perception-based processing. The control task is a pioneering work in the field of deep reinforcement learning. It not only has the perception ability of deep learning but also has the decision-making mechanism in reinforcement learning. van Hasselt et al. [20] innovatively proposed the deep DQN algorithm to improve the problem of overestimating Q value in deep reinforcement learning, which is more accurate in Q value estimation. Subsequent researchers [21] added a recurrent neural network structure to the DQN model, which enabled the model to have time memory capabilities and better process time-series data. At present, the models of deep reinforcement learning are developed in the direction of structural diversification and complex modules. There are many kinds of deep learning methods that can be integrated into reinforcement learning [22, 23]. By adding an attention mechanism to the model, the intelligent physical ability makes more reasonable judgments according to the importance of the system environment and state space, that is, automatic decision-making and tuning.

3. Proposed NFP-WD Model

In the wide model part, we introduce the field-aware factorization machine (FFM) to process the characteristics of conventional network alarm logs. In the deep model part, we use LSTM to process features with time series in device alarm data. The proposed model structure of the NFP-WD is shown in Figure 2.

3.1. Improved Wide Model Based on FFM. The Wide side of the model uses the FFM to deal with a large number of sparse features in the network alarm log data. The linear model of combined features simply considers each feature independently and does not consider the relationship between features. Therefore, we consider using the field-aware factorization machine model FFM to characterize the correlation between features.

There are some sparse features belonging to the same field in the actual data. For example, in the network alarm log data, "alarm level" belongs to a general field feature, which consists of fields such as "prompt," "important," "minor," "urgent," etc. When combining features, we should generalize these sparse features into the same feature field. The field-aware factorization machine can divide the same features into the same field, and the output of the FFM can be expressed as follows:

$$\phi_{\mathrm{FFM}}(w, x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < \mathrm{v}_{i,f_j}, \mathrm{v}_{j,f_i} > x_i x_j. \tag{1}$$

In which, ω_0 is the initial weight, for each one-dimensional feature component x_i, the model automatically learns an implicit vector $\mathbf{V}_{i,f_j}$ for the field f_j where the other feature is located. Using the FFM model as the structure on the wide side can make the model generates multiple independent latent vectors better and learn new warning features. After that, the output of the FFM will be connected to the fully connected layer, and the fully connected layer will extract the cross features generated by the FFM.

3.2. Improved Deep Model Based on LSTM. The deep side of the model adopts the LSTM to train the time series features of the network fault alarm information in the edge environment. LSTM saves the past state information by introducing the unit state c_t, where the forgotten gate f_t determines the content that needs to be forgotten in the unit state, and the input gate i_t determines the content that needs to be newly added to the unit state.The output gate o_t is used to decide whether the cell state c_t will be propagated to the final state h_t. The relevant recursive equations are as follows:

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ f_t &= \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ c_t &= f_t \times c_{t-1} + i_t \times \tanh(W_c[h_{t-1}, x_t] + b_c) \\ h_t &= o_t \times \tanh(c_t), \end{aligned} \tag{2}$$

where i_t, f_t, o_t, c_t, h_t represent the input gate, forget gate, output gate, unit state, and hidden state, respectively, b_i, b_f, b_o, b_c are their corresponding bias terms.σ, tanh represent the sigmoid activation function and the tanh activation function, respectively.

The fully connected layer of the model combines the output of static features after passing through the wide

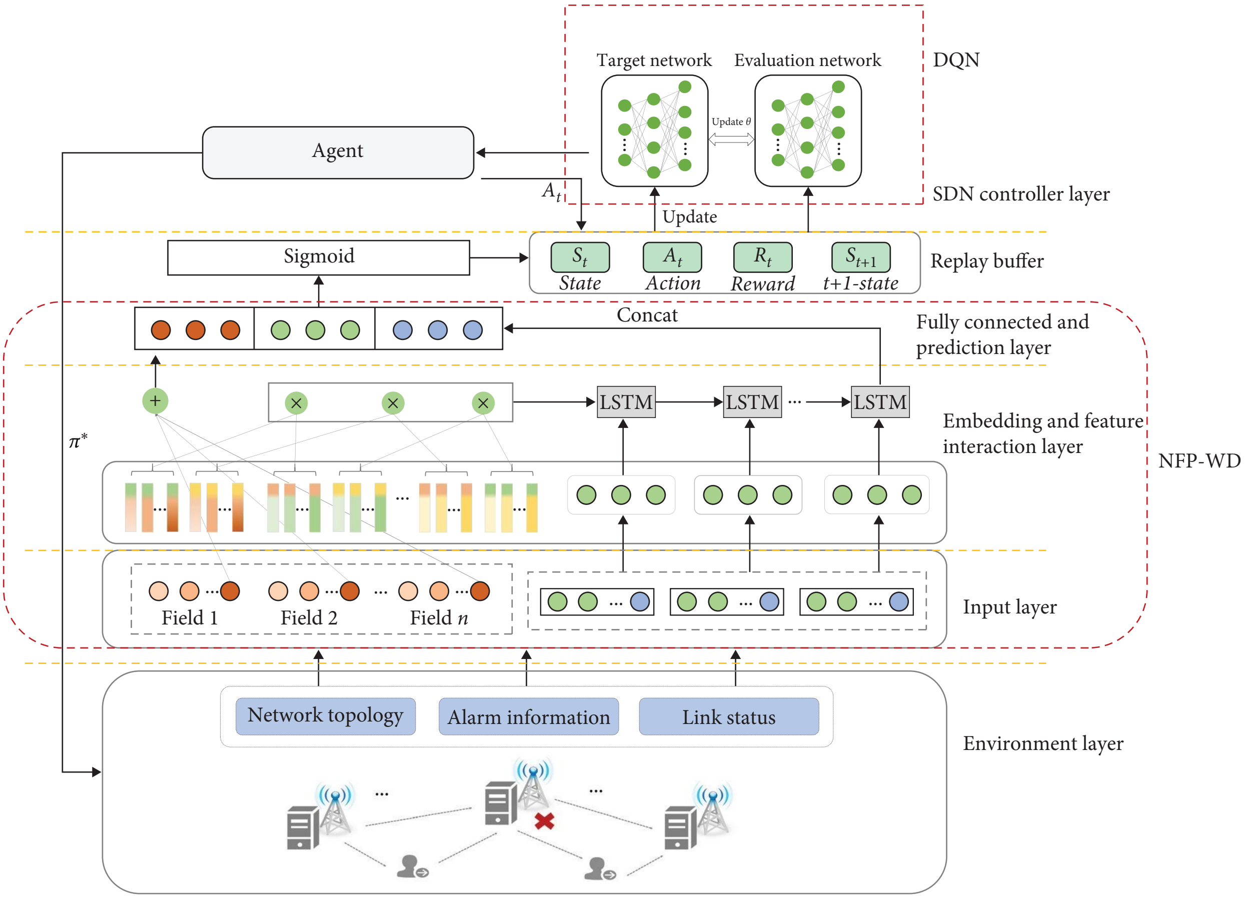


Figure 2: The architecture of SMNFP.

module and the output of dynamic time series features after passing through the deep module and uses the sigmoid activation function to output the probability value of edge network fault prediction in each time window. The output can be formally expressed as follows:

$$y_{\text{con}} = \text{sigmoid}\left(w_{\text{con}} \cdot \text{concat}\left(y_{\text{wide}}, y_{\text{deep}}\right) + b_{\text{con}}\right), \tag{3}$$

where concat is a combination function, which is used to perform vector splicing of the processing output of static nontemporal features and the output of time-series features in each time window. y_{wide} and y_{deep} are the outputs of the FFM and LSTM neural networks, and w_{con} and b_{con} are the weight and bias parameters to be trained.

3.3. Loss Function and Optimization. Our objective function consists of Wide model part and Deep model part. In the Wide part, we use the field-aware factorization machine to cross-feature combinations and generate new alert features. The model uses logistic loss as the loss function and uses the L2 penalty term. To avoid overfitting, L2 penalty term is introduced to penalize the weights of the model, encouraging the model to prefer smaller weight values, thereby reducing model complexity. At the same time, it prompts the model to assign smaller weights to irrelevant or redundant features, improving the model's generalization ability.

The optimized loss function is as follows:

$$L_W = \sum_{p=1}^{N} \log\left(1 + \exp\left(-y_p \phi\left(w, x_p\right)\right)\right) + \frac{\lambda}{2}\|w\|_2^2, \tag{4}$$

where $y_p \in \{0, 1\}$ is the label of the pth sample. λ is the regularization coefficient.

Using the LSTM neural network to predict whether the edge network will fail in the next time period is essentially a binary classification problem, and the loss function can be expressed as follows:

$$L_D = -\sum_{i=1}^{N} y \log \hat{y} + (1 - y) \log\left(1 - \hat{y}\right), \tag{5}$$

where N is the total number of samples, y is the real label of the sample fault, $\hat{y}$ is the probability value that the model predicts the sample to be a positive class value. The NFP-WD model outputs the probability value of network fault in each time window, which leads to corresponding local errors at each step. For the problem of fault prediction, the focus should be the output probability of the model in the last step. Therefore, by adjusting the proportion of the prediction probability of the last step in the global, the object that the loss function should focus on is controlled. The optimized loss function is as follows:

$$L_{D^\circ} = \frac{1}{N}(1-\alpha)L_D + \frac{1}{T}\alpha L_D. \tag{6}$$

Among them, T is the length value of the input sequence, and the hyperparameter $\alpha \in \{0, 1\}$is used to control the importance of the output in the prediction process to the final prediction result. The overall loss function of the final model is as follows:

$$L = L_w + L_{D^\circ}. \tag{7}$$

The goal of NFP-WD model training is to minimize the loss function L. Based on the above design, the Wide model and the Deep model are combined through a fully connected layer, and the final network fault prediction value is obtained after joint training.

4. Proposed Service Migration Method

We first introduce an SDN controller into the mobile edge network and use the controller to monitor the operation of all edge servers, collect all observable computing tasks and network device alarm information, and predict the faults of network equipment in each time window according to the alarm information. When a user moves from one location to another, service migration will be triggered. In order to avoid passing through faulty servers during the migration process, we introduce the NFP-WD module to avoid servers that are about to fail by setting a reasonable reward function; finally, we use deep reinforcement learning to solve the service migration strategy. The overall architecture of the model is shown in Figure 2.

4.1. Service Migration Model Based on Markov Decision Process. We adopt a deep reinforcement learning method to solve the problem of service migration. Our method is based on time windows, that is, each time window t is regarded as a sampling interval, and in each sampling interval, network faults are predicted according to the edge network conditions, and the corresponding service migration decisions are made. Reinforcement learning problems can be formally represented by quintuples of Markov decision processes: $M = (S, A, P, R, \gamma)$, where S is the state space, representing a set of state states, and A is a set of actions, $P(s'|s, a)$ represents the transition probability of taking action a in state s and transitioning to state s$'$. R represents the reward function. γ represents the discounting factor.

4.1.1. State Space S and Action Set A. Suppose the edge network consists of edge servers with N nodes, denoted as $N = (1, 2, \ldots, n)$, the service runs on K servers, the collection of these services is represented as $SE = (se_1, se_2, \ldots, se_k)$. Define a set of nodes $N_f = (f_1, f_2, \ldots, f_n)$ indicates the nodes that will fail after being predicted by the NFP-WD module within a specific time window, during the actual migration process, these nodes will be avoided according to a certain migration strategy. Assume that at a certain moment the user enjoys the service S_{et} provided by the edge node N_t, we define $s(t)$ as the distance between user u at time slot t and the edge server N_t serving it: $s(t) = \|loc_{u_t} - loc_{N_t}\|$, where loc_{u_t} represents the location of user, loc_{N_t} represents the location of edge server. State space $S = \{s(t), t = 1, 2, \ldots, n\}$. In each time window t, the state changes from $s(t)$ to $s'(t)$ after taking action $a(s(t))$. Action set A is the set of these actions $a(s)$, where $a(s(t)) = \begin{cases} 0, & \text{No Service Migration.} \\ 1, & \text{Perform Service Migration.} \end{cases}$

4.1.2. Cost Constraints. We consider the migration cost and communication cost in the process of migrating services from a source server to a target server. Suppose the address of the origin server is l_{ori}, the target server address is l_{dest}, the user address is l_{user}. We measure the distance between two servers by the number of hops between two cellular networks: $\delta = \|l_{\text{ori}} - l_{\text{dest}}\|$, the distance between the user and the target server after the service migration is performed as $\tau = \|l_{\text{user}} - l_{\text{dest}}\|$. We define the migration cost function as $m(\delta) = \begin{cases} \omega_o + \omega_d\theta^\delta\, \delta>0 \\ 0,\ \delta = 0 \end{cases}$. The communication cost function as $\mathrm{n}(\tau) = \begin{cases} \mu_0 + \mu_d\lambda^\tau,\ \tau>0 \\ 0,\ \tau = 0 \end{cases}$, where $\omega_{\text{o}}, \omega_{\text{d}}, \mu_{\text{o}}, \mu_{\text{d}}$, $0 \leq \lambda \leq 1, \theta \geq 1$ are real values. So the total cost function is $C(s, a) = m(\delta) + n(\tau)$.

4.1.3. Reward Function. Suppose that in a certain state s, for a service to be migrated, there is an edge node sequence $N_s = \{N_1, N_2, \ldots, N_{\text{dest}}\}$ representing the migration path of the current service, $N_f = (f_1, f_2, \ldots, f_n)$ represents the set of nodes that may fail predicted by the NFP-WD module in the current state of the system. Define N_K to represent the set of nodes where the service has been deployed. In order to encourage the reinforcement learning mechanism to try to avoid faulty nodes in the migration decision, we define the fault penalty function Penalty(s). The value of the penalty function is determined by whether the faulty node is included in the current migration decision and the origin of the service request. For each state s in the state space S:

$$\text{Penalty}(s) = \sum_{f_{\text{i}} \in N_f} g(f_i, N_{\text{dest}}) + \sum_{n \in (N - N_K)} x_n \operatorname*{dis}_{\mu \in N_K} \{n, \mu\}, \tag{8}$$

where $g(f_i, N_{\text{dest}})$ indicates the number of paths affected by a single failed node f_i, in the network topology, it is expressed as the total number of paths without loops that reach the target node N_{dest} with the faulty node f_i as the starting point. x_n represents the total number of requests at node n. dis $\{n, \mu\}$ represents the shortest distance from node n to the first node of the deployed service. It can be seen that the penalty function is divided into two items. The first item indicates that if there is a faulty node in the migration path, the penalty will be obtained. If a service request is initiated at the edge node where the service is deployed, a penalty will be obtained, and the amount of penalty will also increase with the increase of the number of requests. So the final reward function is as follows:

$$R(s, a) = \text{Penalty } (s) - \text{Penalty } (s') - w_p C(s, a), \tag{9}$$

where $C(s, a)$ represents the cost function. If the state of the system is improved after the migration action is performed, it will receive a positive immediate reward, otherwise it will be punished. The migration strategy needs to strike a balance between the penalty function and the cost function, so we introduce a compromise weight factor w_p to achieve this purpose.

4.2. Service Migration Method Based on Deep Reinforcement Learning. Reinforcement learning is generally used in scenarios that need to interact with the environment. For a given state in the state space, the program selects a corresponding action according to a certain strategy. After the action is executed, the environment changes and the state changes to a new state. After each action is performed, the program will get a reward value, and then the program adjusts its strategy according to the size of the reward value. After all steps are executed, when the program reaches the terminal state, the sum of the rewards obtained is the largest, and the strategy obtained at this time the optimal strategy.

The Q-learning algorithm is a representative algorithm among value-based algorithms in reinforcement learning. Q (s, a) is a state-action value function in reinforcement learning, which represents the sum of the expected total rewards after taking action a in state s. The update process of Q (s, a) is as follows:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right], \tag{10}$$

where $Q(s', a')$ is the expected total return after taking the next action a', $\alpha \in [0, 1]$ is the learning rate used to control the convergence of the model, r represents the reward obtained after taking action a in state s. $\gamma \in [0, 1]$ is discounting factor, which is used to control the degree of influence of the new Q value on the previous Q value.

However, Q-learning uses a Q table to store action values. In the service migration environment we constructed, in order to verify the impact of equipment fault on the migration effect, a large number of edge devices are required, which easily leads to an excessively large state space. Therefore, it is not practical to store the Q value of each time step by constructing a Q table. To solve this problem, we use the Deep Q Network (DQN) algorithm in deep reinforcement learning to calculate the Q value that can be obtained by selecting an action a for given state s. To prevent overfitting, DQN includes an evaluation neural network and a target neural network, which has the same structure but different weight vectors and corresponding biases of the depth neurons. Equation (12) has a similar structure to Equation (11), with the difference that the neuron weight vector in the evaluation network is θ, and the output is $Q(s, a; \theta)$, θ varies with each time step t. The parameters in the target network are the parameters $\hat{\theta}$ in the evaluation network some time ago, and the output is $\hat{Q}(s, a; \hat{\theta})$. After a period of time, the parameters of the evaluation network $\hat{Q}(s, a; \hat{\theta})$ are assigned to the target network. $\gamma \in [0, 1]$ is still discounting factor. The service migration algorithm is described as Algorithm 1. The update process of the action value function can be expressed as follows:

$$\begin{aligned} &\mathrm{Q}(s_t, a_t; \theta) \leftarrow \\ &\mathrm{Q}(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a_{t+1}} \hat{\mathrm{Q}}\left(s_{t+1}, a_{t+1}; \hat{\theta} \right) - \mathrm{Q}(s_t, a_t; \theta) \right]. \end{aligned} \tag{11}$$

In order to solve the problem of unstable training effect caused by the nonindependence of training samples, we use the experience replay buffer as the training method of neural network. Experience replay buffer refers to storing the quadruplets obtained during the training process in the experience pool and then randomly selecting a batch of quadruplets (s_t, a_t, r_t, s_{t+1}) from the experience pool as a batch for training. This random sampling can reduce the correlation between data samples and improve the training efficiency of the neural network. The loss function can be expressed as follows:

$$L(\theta) = E\left[\left(r_t + \gamma \max_{a_{t+1}} \hat{\mathrm{Q}}\left(s_{t+1}, a_{t+1}; \hat{\theta} \right) - \mathrm{Q}(s_t, a_t; \theta) \right)^2 \right]. \tag{12}$$

The execution time of Algorithm 1 is as follows:

$$\begin{aligned} &T(\text{episode } t) = t_0 + t_1 + \\ &(t_2 + t_{2.1} + (t_{2.2} + t_{2.2.1} + t_{2.2.2} + \ldots + t_{2.2.9}) \times T) \times E \\ &= t_{c1} + (t_{c2} + t_{c3} \times T) \times E \\ &= t_{c1} + (t_{c2} \times E + t_{c3} \times T \times E) \\ &= t_{c1} + t_{c2} \times E + t_{c3} \times T \times E \\ &= t_{c3} \times T \times E \\ &= T \times E. \end{aligned} \tag{13}$$

where t_0 represents the time required to initialize the experience pool. t_1 represents the time to initialize the evaluation network and the target network. For each episode, the execution time consumed is t_2, repeatedly running E times. Then, each individual step in Algorithm 1 corresponds to an independent step-by-step time. When the values of episodes and t of the algorithm are very large, the constant terms in T (episode, t) and the coefficients of T and E are negligible. The main influence of the T (episode, t) is not

```
Input:
    State set $S$, Action set $A$, discounting factor $\gamma$, explore probability $\epsilon$
Output:
    Migration strategy $\pi = (0, 1, 2, \ldots T)$.
Initialize the Experience Pool with a capacity of $M$
Initialize the evaluation network neuron weight vector $\theta$
Initialize the target network neuron weight vector $\hat{\theta} = \theta$, the rest of the parameters are the same as the evaluation network
for episode = 1, 2…$E$ do
    Initialize user location $loc_u$ and the location of edge server $loc_N$,initialize the first state $s_1$
    for $t$ = 1, 2…$T$ do
        Predict the faulty node $f$ and add it to the set of faulty nodes $N_f$
        Randomly choose action $a_t$ with probability $\epsilon$
        Or choose the action $a_t = \arg\max(s_t, a_t; \theta)$
        perform action $a_t$,calculate the penalty value p, reward value $r_t$ and the next moment state $s_{t+1}$
        Put the sample $|s_t, a_t, r_t, s_{t+1}|$ into the experience pool
        Randomly select a small batch of samples $|s_j, a_j, r_j, s_{j+1}|$ from EP
        if if episode terminates at step $t+1$ then
            set $y_t = r_t$
        else
            set $y_t = r_t + \gamma \max_{a_{t+1}} \hat{Q}(s_{t+1}; a_{t+1}; \hat{\theta})$
        end if
        Train the network according to the loss function $(y_t - Q(s_t, a_t; \theta))^2$
        Set $\hat{\theta} = \theta$ every $x$ steps
    end for
end for
```

Algorithm 1: Service migration based on network fault prediction and DQN.

E alone but $T \times E$, because $T \times E$ grows much faster than E itself. Therefore, T (episode, t) = O (n_t n_e), where n_e represents the number of episodes and n_t represents the number of time steps in each episode.

5. Experiment

In this section, we will first conduct experiments on the proposed network fault prediction method and compare its prediction effect with the baseline method. Then we apply this method to the service migration process to evaluate the effect of the migration.

5.1. Datasets and Metrics. We select the network alarm log information of the public metropolitan area network from January to November 2013 to train the NFP-WD model. In order to simulate the migration situation when the edge network fails, we refer to the failure information of 10 associated devices in the data set in December 2013 to perform active fault injection on the edge servers.

We use the DeepFace face recognition application as a test program. A cloud server is set up to store the face recognition video dataset iQIYI-VID. The face recognition application is initially deployed in the edge server closest to the user. The mobile user first downloads the face recognition video from the cloud server and then uploads the video to the edge server for recognition. The SDN senses that the user moves to the coverage area of the next edge server. Service migration will be triggered during the migration process, and the face recognition application will be interrupted until after the migration is completed, the user establishes a connection with the new edge server, and the video stream continues.

The alarm log includes the alarm type, alarm severity, alarm name, alarm source, NE type, alarm time, alarm clear time, and confirmation time et, as shown in Table 1.

The alarm levels in the alarm log are divided into four different levels: prompt, minor, important, and urgent. The types of alarms are divided into common alarms and root-cause alarms. In actual situations, only the records with the alarm level of "urgent" in the alarm dataset are defined as fault conditions, and the prediction of network faults is the prediction of emergency-level alarms. Our goal is to predict whether the devices in the network will fail urgently under the conditions of a given time window based on the alarm information.

In order to evaluate the proposed network fault prediction effect, we select Recall, F-Measure, and AUC value as the evaluation indicators of the model.

$$
\begin{aligned}
\text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\
\text{AUC} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \\
\text{F1} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.
\end{aligned}
\tag{14}
$$

where TP indicates the situation that the failure is predicted to occur and the failure occurs, FP indicates the situation that the failure is predicted but does not occur, FN indicates the situation that the failure is predicted not to occur but the failure occurs, and TN indicates that the prediction does not occur and the failure does not actually occur.

In the simulation experiment part of service migration, we select the migration success rate, migration cost, and migration times as the evaluation indicators of the model. Service migration success rate is an important experimental metric in our experiments; we define it as the percentage of how many face recognition applications have completed running properly.

5.2. Baselines and Parameters Settings. First, we compare the proposed network fault prediction method NFP-WD with methods: Bayesian Network [24], Random-Forest [25], SVM [26], Wide&Deep [15], and DeepFM [27]. Afterward, we

Table 1: Statistics of metro network fault dataset.

ID	Alarm type	Alarm level	Alarm name	Alarm source	Positioning information	Time of occurrence
26	Source alarm	Important	Power module power failure warning	Device 1	Entity name = PWR board 8	10/16/2013 17:07:45
1253	Common alarm	Minor	Link down	Device 3	Interface index = 16	01/10/2013 08:07:43
2259	Common alarm	Urgent	Temperature exceeds threshold	Device 12	Entity name = LPU slot 2	02/07/2013 17:09:29
2259	Derived alarm	Prompt	Link down	Device 28	Interface index = 6	02/16/2014 18:53:22

compare the NFP-WD-integrated service migration method SMNFP with the following methods:

(i) ASM [28]: always perform service migration, also known as the greedy migration method. As users move, services are always migrated to the edge server that is closest to the mobile user, and this method tends to lead to large migration costs.

(ii) Mig-RL [8]: A method for service migration based on the Q-learning algorithm in reinforcement learning, which aims to minimize service cost and maximize service quality.

(iii) DSM [29]: A method for modeling service migration as a distance-aware Markov decision process, focusing on the location between mobile users and edge servers.

(iv) SRSM [30]: A service migration method based on fault state-triggered adaptation, which establishes four different fault models for network states and can constrain migration cost, delay, server resource capacity, and bandwidth for different fault conditions.

Edge servers in different geographical locations represent edge nodes. We abstracted ten edge nodes into a Docker container cluster and used the Kubernetes container management platform (K8S) to implement container resource management. Docker containers can be used to conveniently store and migrate resources and provide certain computing power, and the SDN controller can unify the container cluster management through the OpenFlow protocol.

The experimental environment is CPU Intel® Core™ i9-9980XE @3.00 GHz, 128 GB RAM, and two Titan XP graphics cards. The experimental operating system is Ubuntu20.04; we use Tensorflow to implement the algorithm. We conduct simulation experiments on the Mininet network simulation platform. The experimental environment consists of four parts: mobile terminal equipment, edge server, controller, and cloud server. Mobile devices access edge servers through wireless hotspots. In the experiment, the POX controller is selected as the SDN controller. All edge nodes install a local POX controller to collect network topology information and fault conditions, sense, and initiate service migration and schedule migration tasks. Edge nodes run Open vSwitch software to parse the OpenFlow protocol. The global controller acts as a personal PC for equipment fault monitoring and management of local SDN controllers. All parameters of the experiment are shown in Tables 2 and 3.

Table 2: Simulation parameters.

Parameter	Values
Number of ES	5–10
Number of mobile users	5–30
Area	2 km × 2 km
ES coverage radius	300 m
ES overlap coverage radius	45 m
Moving speed of users	5 m/s
Bandwidth for up/down	20–100 Mbps
Latency between ES and users	50 ms
Fault recovery time	20–60 s
Replay memory size	10,000
Learning rate	0.001
Number of episodes	1,800
Compromise factor w_p	0.05
Discounting factor	0.9
Exploration probability	0.1

Table 3: NFP-WD parameters.

Parameter	Values
FFM	Field size : 3 Feature sizes : (64, 64, 64) Embedding size : 4 Dropout shallow : (0.5, 0.5)
LSTM	Hidden size : 3, 512, 256, 128 Num of layers : 3 Epochs : 64 Batchsize : 256 learning rate : 0.003
Joint training	Activation : Sigmoid
Loss function	α : 0.9

5.3. Performance Evaluation. The experiment explores the influence of the prediction time window on the prediction effect and selects the device 4 with more faults in the network fault data set as the research object to predict whether the device will fail in a given time window. We set the time window as 10 min; the experimental results are shown in Table 4; our proposed NFP-WD model outperforms the baseline model in all three metrics.

From Table 3, we have the following observations: when the prediction time window is the same, the traditional

Table 4: Recall, F1, and AUC for predicting whether device 4 will fail when the time window is 10 min.

Methods	Recall	F1	AUC
Bayesian net	0.7345	0.7252	0.6893
SVM	0.8386	0.8294	0.8147
Random forest	0.8771	0.8681	0.8433
Wide&Deep	0.8882	0.8743	0.8521
DeepFM	0.8943	0.8917	0.8696
NFP-WD	0.9116	0.9032	0.8819

machine learning classification model lacks the generalization ability of the model features compared with the three subsequent models integrated with the deep neural network, so the prediction effect is poor. After the DeepFM model replaces the LR part of the Wide&Deep model with FM, it can learn the low-order and high-order feature combinations of the alarm information at the same time without manual feature engineering, which alleviates the sparsity problem in the alarm data set to a certain extent, and the prediction effect is obtained. At the same time, we noticed that there is a lot of time series information in the alarm log in the actual situation, and the DeepFM model does not have the ability to process time series features due to the lack of memory vectors or memory neural units. In order to solve this problem, our NFP-WD model introduces the LSTM neural network in the Deep layer, which enhances the memory of the model; and the introduction of FFM can distinguish the importance of different combined features compared to FM, for example, in the alarm log, the combination of alarm severity and alarm time is an essential feature. After combining these two advantages, the prediction effect of the model for equipment fault has been improved to a certain extent.

All our experiments take the method of controlling variables and study the influence of a certain factor on the experiment when other factors are the same.

We first explore the service interruption time during the service migration. In our experiments, we calculate the service interruption time every 100 episodes and then calculate the average of all interruption times. As shown in Figure 3, the dashed lines represent the average service interruption times of various methods, and our SMNFP method achieves the smallest average service interruption time of 4.2 s, with SRSM, Mig-RL, DSM, and ASM values of 7.4, 15.5, 17.3, and 34.6 s, respectively. This is because Mig-RL, DSM, and ASM work in a fault-free network environment; for the face recognition detection service, if there are servers in the migration planning path that are about to fail in the short term, the face recognition program will be temporarily hung resulting in service interruption until a specific fault recovery time is experienced and the migration process will continue, so the service interruption time for these three methods is longer than the fault-triggered adaptive method SRSM and our SMNFP method.

Figures 4 and 5 show the effect of different numbers of edge servers on the number of service migrations and the average cost. In the scenario of network fault, as the number of servers continues to increase, the state space continues to expand, and the number of hops between the user and the original server that remains connected increases when the user moves quickly within a certain period. To complete the migration goal, the number of migrations and the total cost needs to be increased accordingly. When the number of edge servers is 10, the number of migrations and the average cost of the ASM method are the highest, reaching 272 and 641, respectively, because it is always connected to the server closest to itself and constantly initiates migration requests during the frequent movement of users. The number of migrations and the average cost of the DSM method are 217 and 473, respectively. The number of migrations and the average cost of Mig-RL are 165 and 437, respectively.

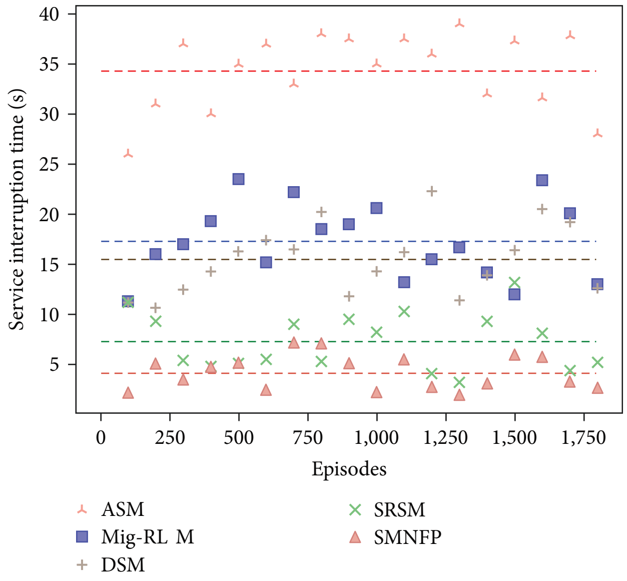


Figure 3: Service interruption time for different methods.

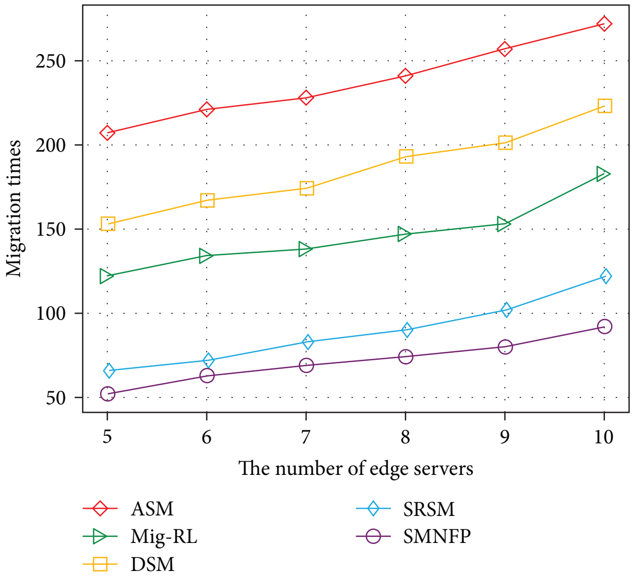


Figure 4: Service migration times under different number of edge servers.

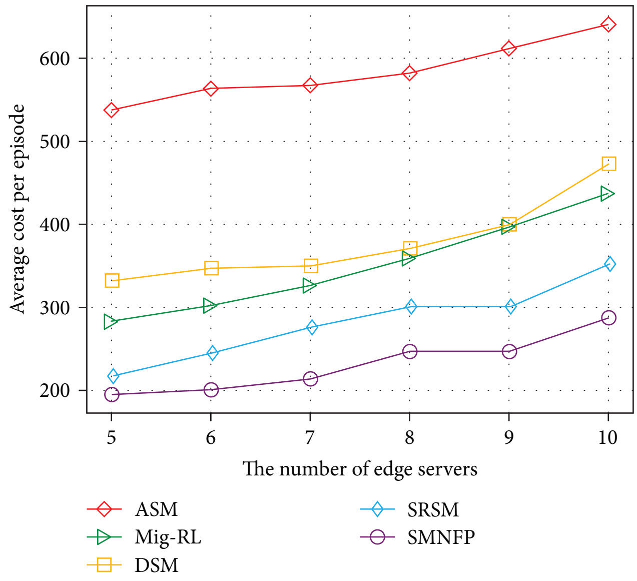


Figure 5: Average cost per episode under different number of edge servers.

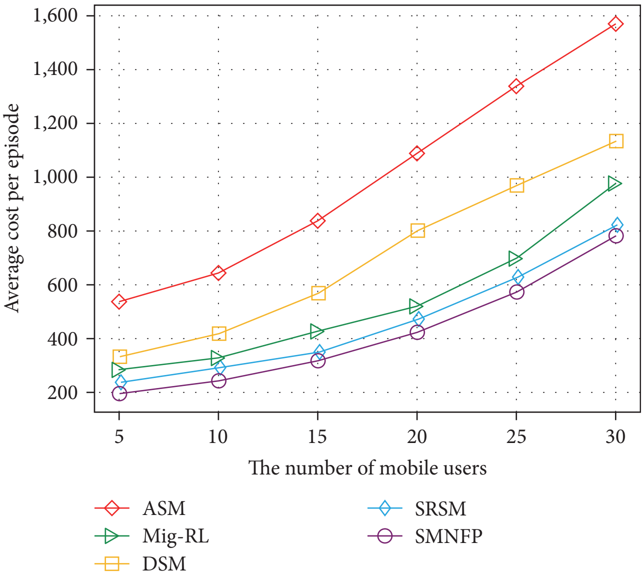


Figure 6: Average cost per episode under different number of mobile users.

SRSM and SMNFP use the DQN to make migration decisions, which can still maintain good performance when the state space increases.

In terms of the number of migrations, the SRSM method and SMNFP method are 127 and 92, respectively, and the average cost is 352 and 287, respectively. The service migration method based on fault adaptation is more complex when faults are frequently triggered and also leads to a high average cost when self-adaptation fails. By predicting and avoiding faulty nodes, our method improves the success rate of migration and reduces the time cost and communication cost of migration as well as the number of migrations.

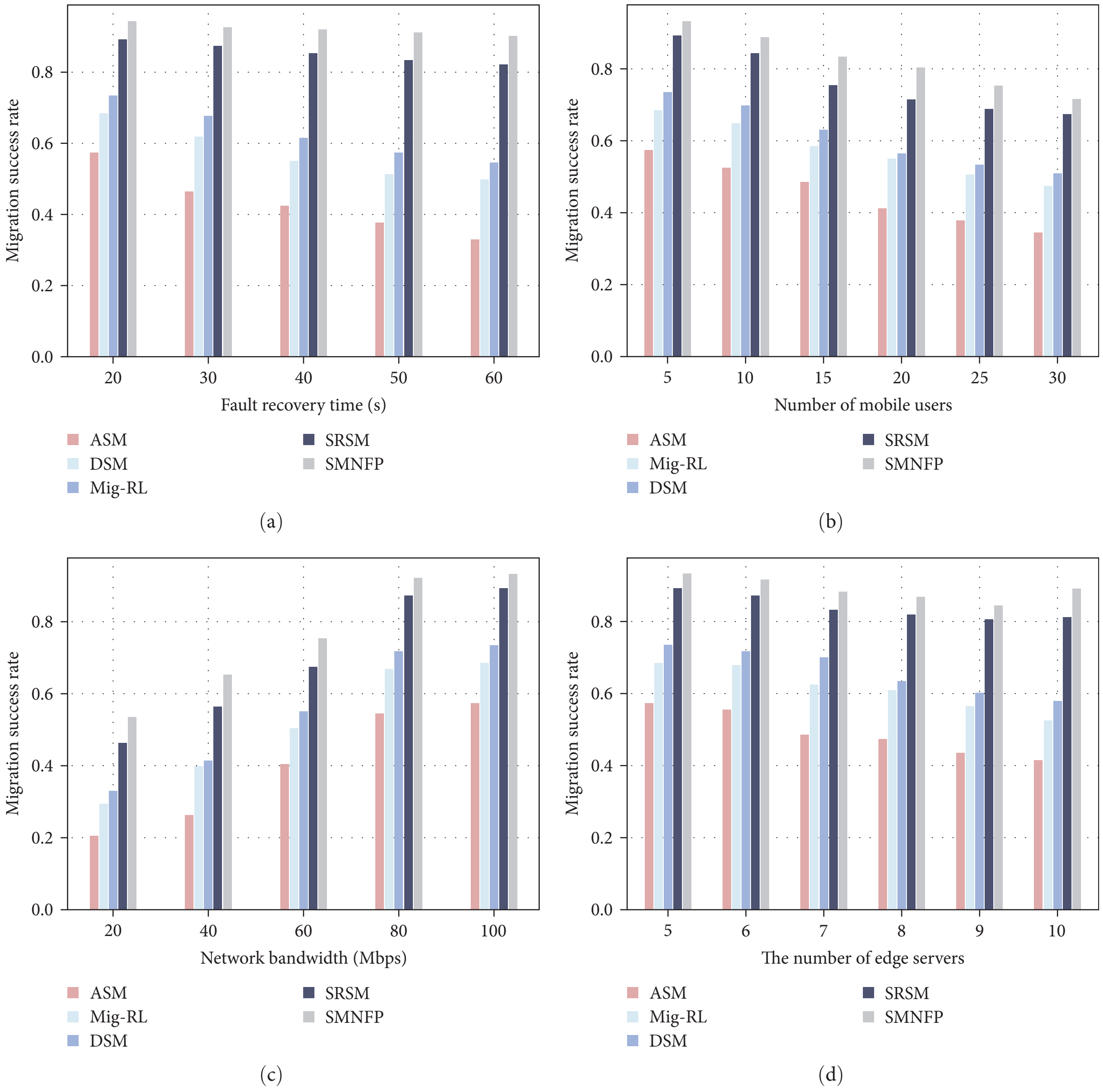
We also study the effect of different numbers of mobile users on migration costs, as shown in Figure 6. Compared with the cost impact of the number of servers, the cost of each method increased significantly when the number of mobile users increased from 5 to 30. This is because as the number of users increases, the total amount of data requested by users increases, and the amount of data that needs to be migrated also increases. Due to the limitation of storage capacity and bandwidth resources, some services cannot be migrated for a short period during the mobile process, resulting in a sharp increase in communication costs. Our proposed SMNFP method has the lowest migration cost among all methods, with a value of 792 when the number of users is 30.

To better simulate the migration situation when a network fault occurs, we choose to periodically clear the fault after a short period of active fault injection into the device. A short fault recovery time can reduce the service interruption time during the migration process, thereby improving the success of the migration rate. When the number of mobile users is five, Figure 7(a) shows the effect of fault recovery time on the migration success rate from 20 to 60 s, and SMNFP achieves the highest migration success rate at different periods. All methods show a decreasing trend as the recovery time increases. When the time is 60 s, the success rate of migration reaches the lowest, which are 0.327, 0.497, 0.544, 0.821, and 0.903, respectively, and the migration success rate of the three methods of ASM, DSM, and Mig-RL decreases greatly. The migration rate of SMNFP has a little downward trend; this is because after NFP-WD predicts the faulty node, only a very small part of the services will be migrated to the faulty node, so the fault recovery time has little effect on SMNFP.

When the fault recovery time is 20 s, we study the effect of the number of concurrent requests on the success rate of migration. The experiment selects different numbers of mobile users from 5 to 25. It can be seen from Figure 7(b) that with the increase in the number of mobile users, different migration methods all showed a significant downward trend. When the number of mobile users is 25, the success rates are 0.343, 0.473, 0.508, 0.642, and 0.714, respectively. This is because, with the increase of users, the number of concurrent face recognition applications increases. Due to the limitation of its bandwidth, computing power, and frequent faults, edge servers have caused a large number of service computing faults and migration faults, and the migration success rate has dropped significantly.

Besides, we study the effect of network bandwidth and the number of edge servers on the success rate of migration. From Figure 7(c), we can see that the success rate of migration decreases with the increase of network bandwidth; when the bandwidth increases from 60 to 80 Mbps, the increase is the highest. When the bandwidth is 100 Mbps, the migration success rate of each method is the highest, which are 0.572, 0.683, 0.733, 0.891, and 0.931, respectively. Continuing to increase bandwidth has little effect on improving the success rate because the factor limiting the success rate of migration is no longer bandwidth but other computer hardware factors.

Figure 7(d) shows the effect of the number of mobile edge servers on the migration success rate. For the added servers, we also follow the previous fault injection method.



(a)

(b)

(c)

(d)

Figure 7: Comparisons of migration success rate for five methods under different parameters: (a) migration success rate under different fault recovery time; (b) migration success rate under different numbers of users; (c) migration success rate under different network bandwidth; (d) migration success rate under different number of edge servers.

It can be seen that with the increase in the number of servers, the migration success rate of various methods decreases slightly. Experiments show that the number of edge servers has a great impact on the number of service migrations but has little impact on the success rate. This may be because the number of servers has not yet reached a very large number in the actual edge network.

6. Conclusion

In this paper, we first propose a network fault prediction method NFP-WD, which is used to predict the fault of the mobile edge network. Then we model the service migration problem as a Markov decision process, and a penalty function is designed to avoid faulty nodes during migration. Simulation experiments on service migration show that our proposed SMNFP method outperforms several baseline methods.

In the follow-up work, we plan to analyze the user's movement trajectory and predict their movement patterns under network fault scenarios to further improve the success rate of migration.

Data Availability

The original dataset used in this work is available from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62272243), Jiangsu Key Laboratory of Big Data Security & Intelligent Processing and Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] L. Wang, Y. Zhang, and S. Chen, "Computation offloading via Sinkhorn's matrix scaling for edge services," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8097–8106, 2021.

[3] M. Sun, Z. Zhou, X. Xue, and W. Gaaloul, "Migration-based service allocation optimization in dynamic IoT networks," in *International Conference on Service-Oriented Computing*, pp. 385–399, Springer, Cham, 2021.

[4] S. W. Park, A. Boukerche, and S. Guan, "A novel deep reinforcement learning based service migration model for mobile edge computing," in *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pp. 1–8, IEEE, Prague, Czech Republic, 2020.

[5] L. Yin, P. Li, and J. Luo, "Smart contract service migration mechanism based on container in edge computing," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 157–166, 2021.

[6] I. Labriji, F. Meneghello, D. Cecchinato et al., "Mobility aware and dynamic migration of mec services for the internet of vehicles," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 570–584, 2021.

[7] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow-me cloud: when cloud services follow mobile users," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 369–382, 2019.

[8] Y. Wang, S. Cao, H. Ren et al., "Towards cost-effective service migration in mobile edge: a Q-learning approach," *Journal of Parallel and Distributed Computing*, vol. 146, pp. 175–188, 2020.

[9] Z. Liang, Y. Liu, T.-M. Lok, and K. Huang, "Multi-cell mobile edge computing: joint service migration and resource allocation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5898–5912, 2021.

[10] P. Wang, T. Ouyang, G. Liao, J. Gong, S. Yu, and X. Chen, "Edge intelligence in motion: mobility-aware dynamic DNN inference service migration with downtime in mobile edge computing," *Journal of Systems Architecture*, vol. 130, Article ID 102664, 2022.

[11] M. Xu, Q. Zhou, H. Wu, W. Lin, K. Ye, and C. Xu, "PDMA: probabilistic service migration approach for delay-aware and mobility-aware mobile edge computing," *Software: Practice and Experience*, vol. 52, no. 2, pp. 394–414, 2022.

[12] Y. Miao, F. Lyu, F. Wu et al., "Mobility-aware service migration for seamless provision: a reinforcement learning approach," in *ICC 2022-IEEE International Conference on Communications*, pp. 5064–5069, IEEE, Seoul, Korea, 2022.

[13] R. Lin, Z. Pei, Z. Ye, B. Wu, and G. Yang, "A voted based random forests algorithm for smart grid distribution network faults prediction," *Enterprise Information Systems*, vol. 14, no. 4, pp. 496–514, 2020.

[14] S. A. Yadwad and V. K. Vatsavayi, "Fault prediction for network devices using service outage prediction model," *Journal of Communications*, vol. 17, no. 5, pp. 339–349, 2022.

[15] H.-T. Cheng, L. Koc, J. Harmsen et al., "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 7–10, Association for Computing Machinery, New York, NY, United States, 2016.

[16] J. Jia, C. Feng, T. Zhang et al., "Deep fault prediction with flexible weighted mining based alarm correlation analysis of communication networks," in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pp. 173–177, IEEE, Nanning, China, 2020.

[17] P. Klein, N. Weingarz, and R. Bergmann, "Using expert knowledge for masking irrelevant data streams in siamese networks for the detection and prediction of faults," in *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, IEEE, Shenzhen, China, 2021.

[18] Y. Y. Tefera, T. Kibatu, B. S. Shawel, and D. H. Woldegebreal, "Recurrent neural network-based base transceiver station power supply system failure prediction," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, Glasgow, UK, 2020.

[19] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[20] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[21] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," 2015 aaai fall symposium series, 2015.

[22] J.-K. Ge, Y.-F. Chai, and Y.-P. Chai, "WATuning: a workload-aware tuning system with attention-based deep reinforcement learning," *Journal of Computer Science and Technology*, vol. 36, pp. 741–761, 2021.

[23] J. Li, L. Xin, Z. Cao, A. Lim, W. Song, and J. Zhang, "Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2306–2315, 2022.

[24] K. Dejaeger, T. Verbraken, and B. Baesens, "Toward comprehensible software fault prediction models using Bayesian network classifiers," *IEEE Transactions on Software Engineering*, vol. 39, no. 2, pp. 237–257, 2013.

[25] J. Shen, J. Wan, S.-J. Lim, and L. Yu, "Random-forest-based failure prediction for hard disk drives," *International Journal of Distributed Sensor Networks*, vol. 14, no. 11, pp. 1–15, 2018.

[26] H. Zhang, Y. Liu, and L. Wang, "An intelligent alarm method for optical fiber network based on backtracking single alarm information," in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, pp. 56–60, IEEE, Shanghai, China, 2020.

[27] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: a factorization-machine based neural network for ctr prediction," arXiv preprint arXiv:1703.04247, 2017.

[28] L. Ma, S. Yi, and Q. Li, "Efficient service handoff across edge servers via docker container migration," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–13, Association for Computing Machinery, New York, NY, United States, 2017.

[29] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, “Dynamic service migration in mobile edge computing based on Markov decision process,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1272–1288, 2019.

[30] L. L. Rui, M. Zhang, Z. Gao, X. Qiu, Z. Wang, and A. Xiong, “Service migration in multi-access edge computing: a joint state adaptation and reinforcement learning mechanism,” *Journal of Network and Computer Applications*, vol. 183-184, Article ID 103058, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 4758451, 8 pages
https://doi.org/10.1155/2023/4758451

Research Article

Intelligent Construction of Hospital Management Organization Based on Communication Technology and Information Fusion

Wenyan Zhang,[1] Xiujie Chen,[1] Yan Zhang ,[2] and Hao Hua [1]

[1]*Department of Medical Management, Beijing Tsinghua Changgung Hospital, Beijing 102218, China*
[2]*Department of Medical Services, Beijing Tsinghua Changgung Hospital, Beijing 102218, China*

Correspondence should be addressed to Yan Zhang; zya00581@btch.edu.cn and Hao Hua; hha00212@btch.edu.cn

Received 6 September 2022; Revised 24 September 2022; Accepted 7 October 2022; Published 5 May 2023

Academic Editor: Peiying Zhang

With the mature application of 5G communication and the development of artificial intelligence, the deep integration of modern hospital management and information technology has been fully realized. Therefore, this paper explores the organizational structure and system design of our institute and implements the construction and operation of the information system according to the concept of "management institutionalization, organization informatization, and form computerization." The construction of information management departments was strengthened, the entire 69 information systems with systematic thinking were managed, the dynamic management mechanism of information system operation and maintenance was established, and the fine closed-loop management of hospital processes was realized. The results show that the information system based on institutional management will improve the management efficiency of the hospital and ensure the real-time, accuracy, and security of hospital data information.

1. Introduction

In July 2017, the General Office of the State Council issued the guiding opinions on the establishment of modern hospital management system (GCFA (2017) No. 67) [1], in which the information management system is regarded as one of the eight core systems for the construction of hospital internal governance system.

A smart hospital includes three dimensions: smart medical care, smart service, and smart management [2]—"smart medical care" for medical staff, "smart service" for patients, and "smart management" for management [3]. Smart management is mainly the information application level of hospital comprehensive management (human, financial, material, etc.) [4].

Modern hospital management requires the all-round intervention of information technology and follows the law of the "Michel model" (the Michel model is a classic model for judging the degree of informatization and the development stage of informatization; the structure is shown in Figure 1 [5]). The early stage of single-machine and single-user data processing has developed to the stage of department-level management information systems, and now, based on the stage of hospital-level integrated systems and technologies [6], the information systems of each hospital are gradually covering the entire hospital management process. Provide an effective basis for hospital management and decision-making and guide the refined and high-quality development of the hospital, refer to Figure 1 for further details.

Beijing Tsinghua Changgung Hospital (hereinafter referred to as "the hospital") is a large-scale comprehensive public hospital jointly managed by the Tsinghua University and Beijing Municipal Government. Formosa Plastics and Taiwan Chang Gung Memorial Hospital have donated and assisted in the construction and operation of the hospital. At the beginning of its establishment, Taiwan Changgung Hospital's management philosophy was learned, the deep integration of the management system and information system was explored, information of the hospital in

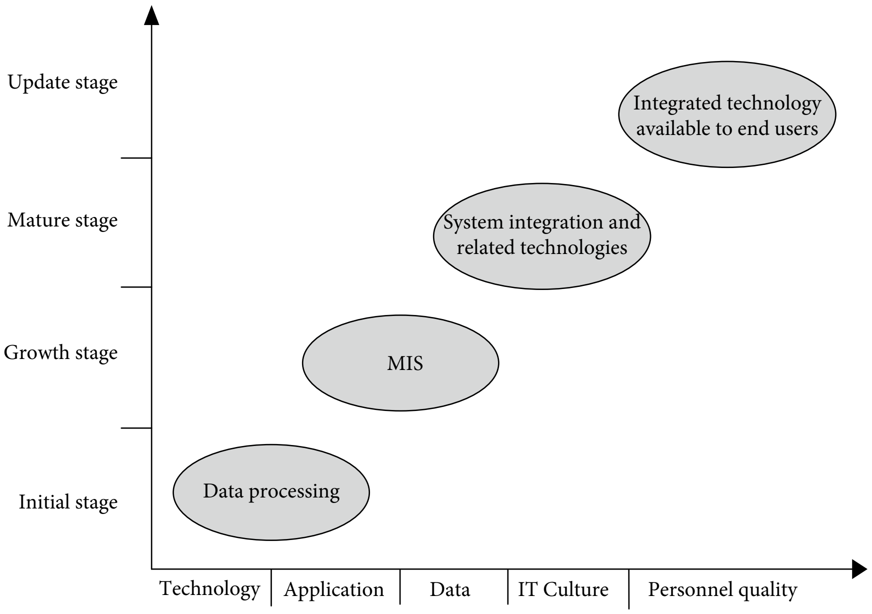


Figure 1: The Michel model.

terms of hospital management informatization practice was highly integrated and shared, and certain experience was accumulated.

The rest of the composition is as follows: Section 2 introduces the organizational structure and institutional guarantee, Section 3 shows the information system operation and management practice, Section 4 describes the modification of the information system, Section 5 is the data query business, Section 6 is the closed-loop management mechanism, Section 7 is the discussion, and Section 8 is the conclusion.

2. Organizational Structure and Institutional Guarantee

2.1. Decentralized Audit. The hospital draws on and localizes the management model of the Chang Gung Memorial Hospital in Taiwan, builds a modern hospital management system and operation model, implements the president responsibility system under the leadership of the party committee, and is managed by a professional medical team and a professional administrative team efficiency and effectiveness. The hospital has set up 33 committees, 15 administrative departments, and 19 business departments. The committee and each administrative department, as the professional management and decision-making team of the president, respectively, assist in the review of medical professional affairs and management professional practice and at the same time realize the separation of the management department and on-site execution, promote the participation of professionals in hospital management; promote scientific, democratic, and effective decision-making; and promote the high-quality development of hospitals [7], refer to Figure 2 for further details.

2.2. Hospital Management Philosophy. Based on the management concept of "institutionalized management, form-based system, and computerized form," the hospital fully realizes the deep integration of the management system and information system and forms a high degree of integration and sharing of information throughout the hospital.

2.2.1. Institutionalization of Management. According to the level and scope of use of the system, the hospital is divided into 74 first-level rules (systems, rules, and regulations), 214 second-level rules (measures), and 412 third-level rules (detailed rules and work points), a total of 700 items. Levels of system examination and approval authority are different. For example, the rules and regulations are formulated (revised) by the system management department, and after being reviewed by the corresponding committee, they are reviewed and approved by the president's office; rules and regulations are reviewed and approved by the party committee.

2.2.2. Formalization of the System. The form mainly reflects the process of business processing based on the system, so that the system can be implemented. The elements of the form include business processing links, role permissions, and opinions.

2.2.3. Form Computerization. The elements in the form are presented in an informative form. The use of information technology to build a refined quality control management platform can improve the work efficiency of the management department [8]. The information system is closely integrated with business processes to support and control the entire process of medical activities, ensuring safe and efficient medical services, and at the same time for continuous improvement. Provide decision support for hospital management and medical services.

2.3. Information Management Department Settings. The hospital has a total of 69 information systems, all of which are managed by 14 corresponding system management

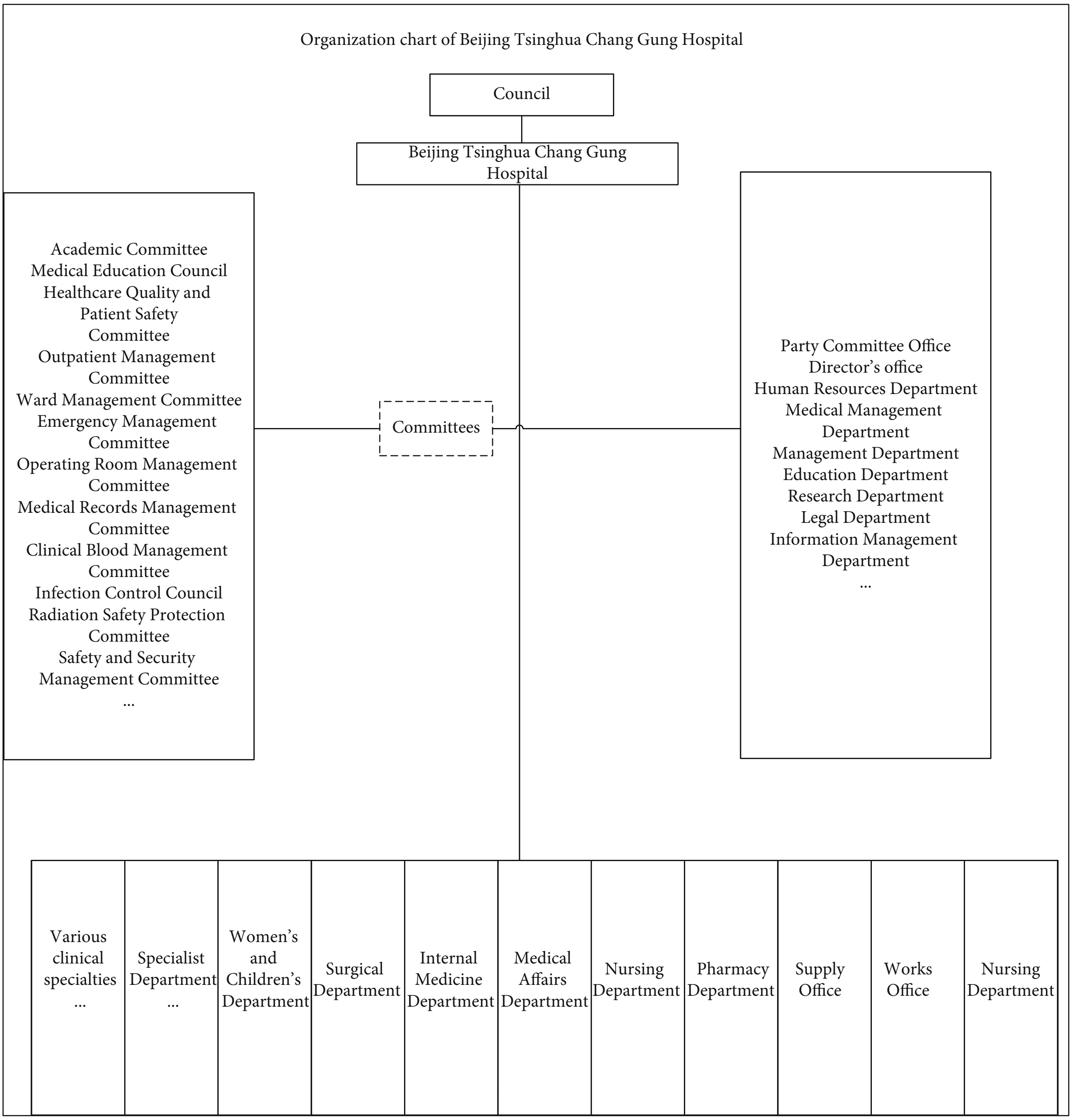

Organization chart of Beijing Tsinghua Chang Gung Hospital
Council
Beijing Tsinghua Chang Gung Hospital
Academic Committee
Medical Education Council
Healthcare Quality and Patient Safety Committee
Outpatient Management Committee
Ward Management Committee
Emergency Management Committee
Operating Room Management Committee
Medical Records Management Committee
Clinical Blood Management Committee
Infection Control Council
Radiation Safety Protection Committee
Safety and Security Management Committee
...
Committees
Party Committee Office
Director's office
Human Resources Department
Medical Management Department
Management Department
Education Department
Research Department
Legal Department
Information Management Department
...
Various clinical specialties ...
Specialist Department ...
Women's and Children's Department
Surgical Department
Internal Medicine Department
Medical Affairs Department
Nursing Department
Pharmacy Department
Supply Office
Works Office
Nursing Department

Figure 2: Organizational chart of the Beijing Tsinghua Changgung Hospital.

departments, namely, the functional department, and the software development team is the entrusted executive department. The system management department designates a system administrator for each system module. The administrator's responsibilities include (1) planning and optimization of the system architecture, (2) setting and assigning system permissions, (3) reviewing system modification requirements, and (4) system documentation renewal. For example, the administrator of the registration system is the specialist in charge of the consultation business of the medical management department, the administrator of the inpatient doctor order system is the specialist in charge of the inpatient service of the medical management department, and the administrator of the personnel system is the management specialist of the human resources department.

2.4. Information Support Department Setup. The hospital has its own information management and technical team, independently develops the hospital information system, fully realizes the deep integration of the management system and the information system, and highly integrates and shares the information of the whole hospital [9]. The information team is divided into a software development group and a hardware maintenance group.

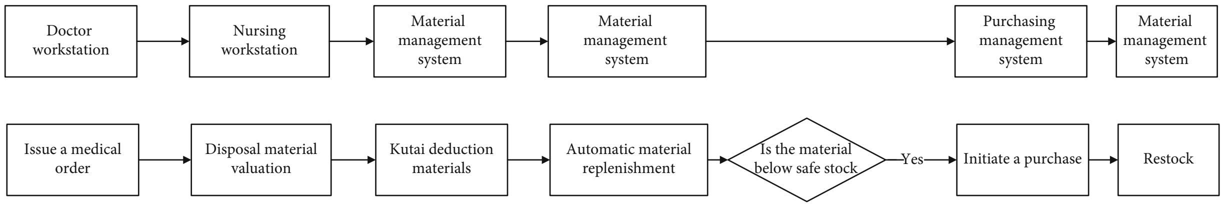

Doctor workstation
Nursing workstation
Material management system
Material management system
Purchasing management system
Material management system
Issue a medical order
Disposal material valuation
Kutai deduction materials
Automatic material replenishment
Is the material below safe stock
Yes
Initiate a purchase
Restock

Figure 3: Example of data sharing across systems.

The information management department and the information support department cooperate to establish an efficient information system for continuous improvement. From the perspective of overall planning, it is better to ensure that the information system can support the improvement of the overall level of the hospital, ensure the realization of teaching and scientific research goals, and have good scalability.

3. Information System Operation and Management Practices

The hospital currently has 69 information systems covering the whole process, mainly including outpatient registration, inpatient charges, emergency doctor workstation, medical record management, and other systems based on the electronic medical record system, and budget, personnel, materials, finance, equipment based on the logistics management system, scientific research, and other systems can achieve a high degree of data sharing across systems. For example, when a doctor issues a treatment order on the resident doctor's workstation, the nursing workstation automatically charges the treatment materials, the cost warehouse of the patient's nursing station automatically deducts one unit of the priced consumables, and the main hospital library automatically replenishes one unit of the priced consumables to the nursing care station. On the station warehouse platform, when the remaining amount of the consumables in the main warehouse is lower than the safety stock set by the system, the system automatically initiates procurement and replenishes the warehouse to achieve full-process control, refer to Figure 3 for further details.

Each system has a set of system management documents, which are maintained and updated by the administrator. Specifically, these include the following:

3.1. System Association Diagram. Show the relationship between the system and other systems, such as the doctor workstation system and the ward nurse workstation, blood management, medical record management, human resources, inspection, hospitalization, and discharge systems. The associated interfaces between systems have numbers and standard descriptions.

3.2. Job Association Diagram. Show the relationship between different modules of the same system, such as the query of historical medical advice information, view of written medical records, query of inspection and test results report, and query of drug information in the doctor workstation system. The associated interfaces between modules are numbered and have a standard description, refer to Figure 4 for further details.

3.3. Transaction Flow Chart. The system is to provide support for each specific business process, each business unit has the transaction process, and the flow chart of the transaction of important two dimensions, respectively, corresponds to the movement and the role of each node, and at the same time shows the instructions and may trigger the form, judgment, control, etc., and flow chart of admission to handle affairs, for example, transaction flow diagram contains the content as follows, refer to Figure 5 for further details.

3.4. Operating Instructions. Each operation screen of all systems has instructions for use, including the format requirements and sequence of input in specific fields, especially the control points. For example, in the hospitalization management system, it is specified that physicians should issue medical treatment to patients who are determined by medical diagnosis within the scope of practice and need to be hospitalized. For admission notice, if a resident physician or a physician has not yet obtained a practicing qualification or has not been given the right to exercise the corresponding specialty duties, the system will not grant admission permission, and the physician control cannot issue an admission notice.

The following are the listed tips:

(1) *Form Type*. Hospital admission notice issued

(2) *Function*. Issue admission notice

(3) *Use time*. To issue admission notice

(4) *Department of Use*. Outpatient physicians

(5) Usage

Click the "Appointment Management"—"Opening hospital Notice" button to jump to "Screen 4-2":

(1) Information of the attending physician and department is brought into the system by default, and doctors can click the corresponding field to modify

(2) Patient and diagnosis information is brought into the system by default

(3) The physician selects the date of hospitalization, priority registration, alternative date, scheduled

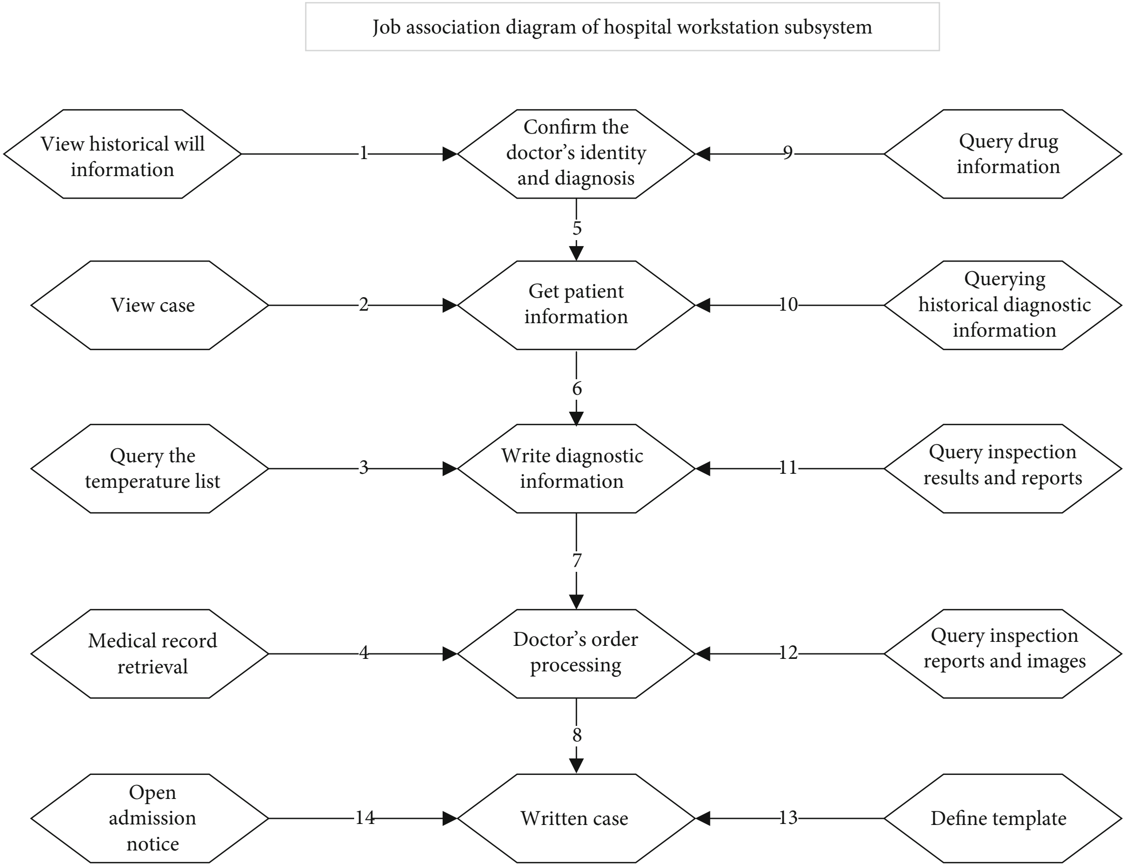

Job association diagram of hospital workstation subsystem
View historical will information
1
Confirm the doctor's identity and diagnosis
9
Query drug information
5
View case
2
Get patient information
10
Querying historical diagnostic information
6
Query the temperature list
3
Write diagnostic information
11
Query inspection results and reports
7
Medical record retrieval
4
Doctor's order processing
12
Query inspection reports and images
8
Open admission notice
14
Written case
13
Define template

Figure 4: Example of job association diagram.

operation time, advance payment amount, whether special needs patients, whether daytime chemotherapy, whether trauma patients, and whether 30-day readmission

(4) Click "Apply" to complete the issuance of the hospitalization notice, and the system prints the hospitalization notice by default and delivers it to the patient

(5) Click "Reprint notice" to reprint the notice of hospitalization

(6) Click "Cancel pre-hospitalization" to cancel the notice of hospitalization

4. Modification of the Information System

4.1. Timing of Information System Modification and Change

(1) Changes in government orders and regulations such as medical reform

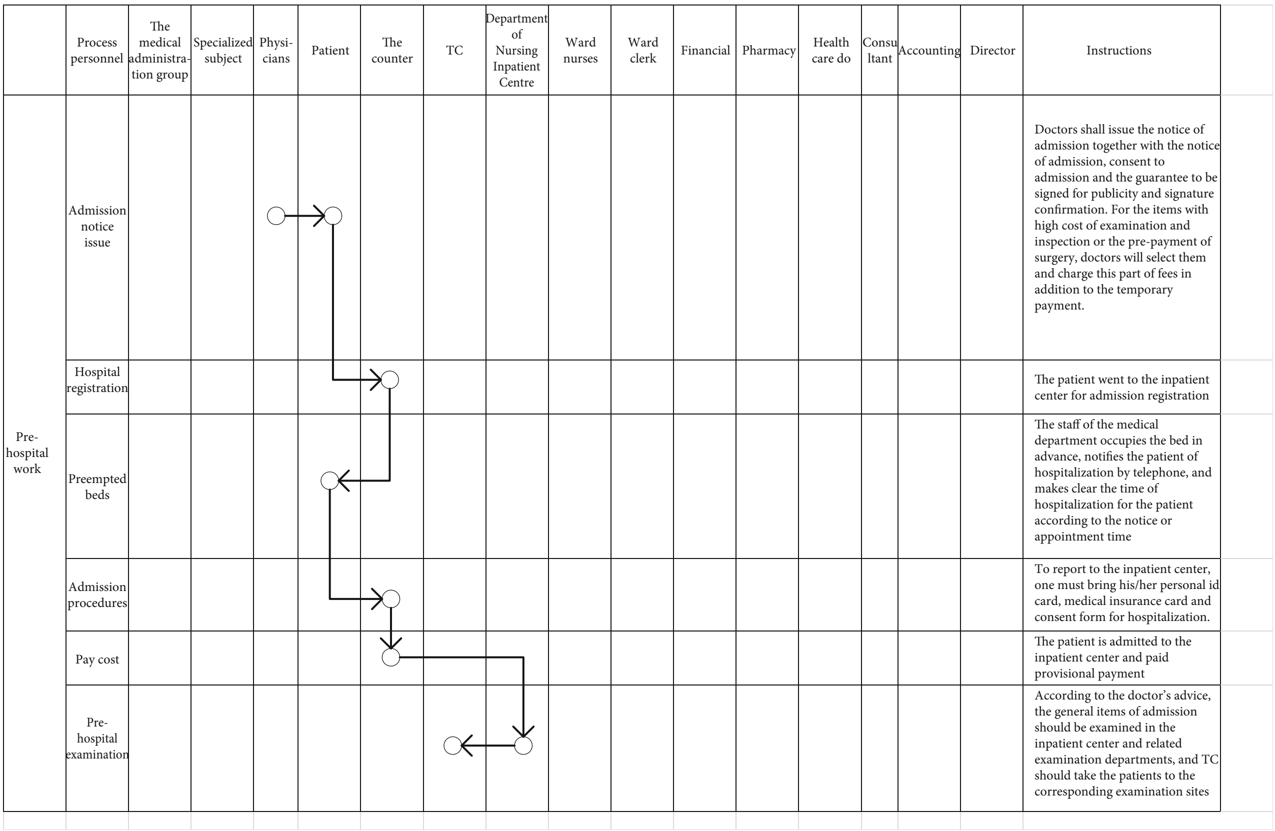
(2) Revision of the hospital system and the discovery of abnormal loopholes in the business process that need to be improved

(3) The constant operation of on-site business personnel affects work efficiency etc.

4.2. Computer Feedback Form. System modifications are handled in the form of computer feedback sheets. The specific process is the following: the user department puts forward the demand and the system management department reviews it. The main point of the review is to first judge whether the demand is consistent with the current system. If it is consistent, the review will continue. The business functions involved will be reviewed/countersigned/organized at a meeting by a single functional department, and then, a summary opinion will be given and submitted to the hospital leadership for verification. If you agree to the modification, the computer opinion sheet will be sent to the information department for modification. After the modification is completed, it will be tested by the user department, and finally, the new function will be launched. From this, it can be seen that the information department is the entrusted execution department, and the core of system modification is to clarify the requirements and review the rationality of the requirements, refer to Figure 6 for further details.

5. Data Query Business

In addition to the regular system operation modification through the computer feedback form, another embodiment of the informatization intervention is to provide an effective basis for hospital management and decision-making



Doctors shall issue the notice of admission together with the notice of admission, consent to admission and the guarantee to be signed for publicity and signature confirmation. For the items with high cost of examination and inspection or the pre-payment of surgery, doctors will select them and charge this part of fees in addition to the temporary payment.

The patient went to the inpatient center for admission registration

The staff of the medical department occupies the bed in advance, notifies the patient of hospitalization by telephone, and makes clear the time of hospitalization for the patient according to the notice or appointment time

To report to the inpatient center, one must bring his/her personal id card, medical insurance card and consent form for hospitalization.

The patient is admitted to the inpatient center and paid provisional payment

According to the doctor's advice, the general items of admission should be examined in the inpatient center and related examination departments, and TC should take the patients to the corresponding examination sites

Figure 5: Example of transaction flowchart.

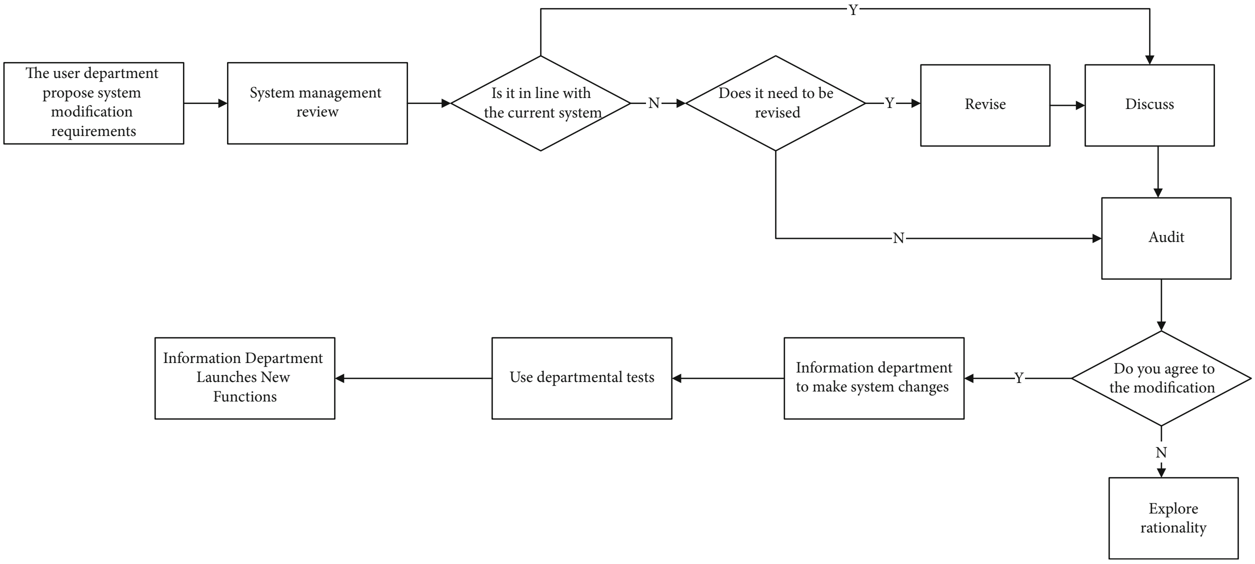

The user department propose system modification requirements
System management review
Is it in line with the current system
Y
N
Does it need to be revised
Y
N
Revise
Discuss
Audit
Do you agree to the modification
Y
N
Information department to make system changes
Use departmental tests
Information Department Launches New Functions
Explore rationality

Figure 6: Processing flow of computer feedback.

through consistent data information covering the entire process. Data query can be divided into routine data query (such as monthly, quarterly, and yearly) and one-time data query according to different business types.

5.1. The Routine Data Query Business Conducts Business Flow through the Indicator Setting Table. The index data that can be queried in real-time provides the hospital with real-time monitoring of quality and efficiency and decision-making reference. For the index that continuously exceeds the threshold, the hospital will focus on reviewing the business operation. For example, on the average length of hospital stays, the indicator is defined as the average length of stay of all discharged patients in a certain period, and the calculation formula is the total bed days occupied by discharged patients/the number of discharged patients in the same period. The numerator is the total number of bed days occupied by discharged patients in each natural month, and the denominator is the total number of discharged patients in each natural month. The threshold is set by the medical management department, and the information department is built into the system. The system can automatically capture the average hospitalization days of the whole hospital and each department within any time range for a specialist and related departments to query and prompt whether the indicator meets the standard according to the threshold in the indicator setting table. The medical management department is based on the data provided monthly by the system. Carry out a subdiscipline review and supervise the improvement.

5.2. The One-Time Data Query Business Conducts Business Flow through the Data Query Application Form. The inquiry department needs to fill in the purpose of data use, data connotation, and statistical time interval, emphasizing that after signing the data use confidentiality commitment, it will be provided by the information department after being reviewed by the system department and the hospital-level supervisor. In the process of system construction, according to the post setting requirements of different departments, from the perspective of information security, the post responsibilities of each department should be clarified, and information security should be ensured to the greatest extent on the basis of strengthening information exchange between departments [10].

6. Closed-Loop Management Mechanism

6.1. Unify the Data Caliber of Each System. The improvement process of the whole information system mentioned above is a self-consistent cycle process of implementing hospital management institutionalization, system form, and form computerized management concepts. The association between them is conducive to monitoring information at each node of the business process, which is easy to find loopholes, and realizes the closed-loop management of institutional processes in the hospital.

6.2. System Revision and System Update Synchronization. When the various systems in the hospital are revised, the relevant information systems should be revised accordingly. When the information system needs to be revised, it should be judged whether it is consistent with the system. After the system is revised, the system management documents should be updated in time to ensure that the system and implementation are always consistent.

6.3. Data Analysis Serves Business Decisions. The process of hospital informatization construction is the reconstruction and transformation of management concepts and models [11]. The information and data of the hospital come from the business, then return to the business, and continuously provide a reference for business decision-making through the comprehensive analysis of the data. Use the information platform to implement intensive and scientific management of each management unit of the hospital, “patient-centered”, with limited investment in medical resources to create more medical service value [12].

7. Discuss

7.1. Formation of the Management Mechanism of “Requirements-Review-Execution” of the Information System. The hospital has strengthened the setting of the information management department, which is based on the premise of the decentralization of the administrative function department and the business department. The demand and use of the information system come from and go to the business, and the whole hospital needs to form a consensus on “the business department raises the demand, the functional department reviews and manages, and the information department executes.” The construction of a hospital information system is not only a matter of the information department, nor is it only a matter of requirements without review. Only by systematically managing “requirements-review-execution” can the controllability, consistency, and efficiency of information system construction and management be ensured.

7.2. Informatization Construction and Standardization of Data Specifications Are the Basis for Interconnection. Under the wave of the country’s vigorous development of smart medical care and the strengthening of Internet hospital construction, the connotation and extension of hospital informatization have undergone great changes. The coding and interfaces used between the existing information system of the hospital and the external system have not yet been unified [13–16]. On the premise of ensuring information security, the hospital needs to strengthen the overall planning of information management, strengthen the ability to communicate with external information, and have certain standardized parameter configuration and good scalability without violating the existing management system (the needs of future hospital development). Through standardization construction, the standard compliance of electronic medical record data and shared documents can be improved, and interconnection and business collaboration can be realized [17, 18].

8. Conclusions and Future Work

A modern hospital must have a modern governance system and management capabilities. The realization of “scientific management, efficient operation, and powerful supervision” mentioned in this goal needs to be built on a powerful hospital information system. When the integration level of the hospital information system is higher, the refined management level of the hospital is also higher. The application of information technology and the effective operation of the smart management system have provided new ideas and new models for public hospitals that are in urgent need of improving efficiency and making breakthroughs and innovations [19]. The information system based on institutional management will provide strong support for the hospital to improve management efficiency and guide the high-quality development of the hospital. In the future, we will add supervision modules to make the system more efficient.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

References

[1] E. Schulmann and M. Galeotti, "A tale of two councils: the changing roles of the security and state councils during the transformation period of modern Russian politics," *Post-Soviet Affairs*, vol. 37, no. 5, pp. 453–469, 2021.

[2] X. Xie, X. Pan, W. Zhang, and J. An, "A context hierarchical integrated network for medical image segmentation," *Computers and Electrical Engineering*, vol. 101, article 108029, 2022.

[3] T. Q. Sun and R. Medaglia, "Mapping the challenges of Artificial Intelligence in the public sector: Evidence from public healthcare," *Government Information Quarterly*, vol. 36, no. 2, pp. 368–383, 2019.

[4] D. Yuantai, H. Shuang, and X. Yang, "Design and practice of intelligent hospital information system technology architecture," *Chinese journal of health information management*, vol. 17, no. 6, article 697-701+720, 2020.

[5] L. Minfeng and O. Wenjie, "Research on government information resource management mode under the background of digital economy," *Fintech Times*, vol. 30, no. 1, pp. 28–39, 2021.

[6] M. Chen, W. Tao, L. Xianfeng, and X. Weiguo, "Construction of integrated hospital information platform," *China hospital*, vol. 15, no. 4, pp. 61–64, 2011.

[7] X. Xie, X. Pan, F. Shao, W. Zhang, and J. An, "MCI-net: multiscale context integrated network for liver CT image segmentation," *Computers and Electrical Engineering*, vol. 101, article 108085, 2022.

[8] T. Liu, "The application of machine learning models in network protocol vulnerability mining," *Security and Communication Networks*, M. A. Khan, Ed., vol. 2022, Article ID 9086938, pp. 1–8, 2022.

[9] Z. Yuehong and L. Yuqian, "Exploration and practice of medical management division and collaborative operation management mode in Beijing Tsinghua Changgung Hospital," *Health resources in China*, vol. 24, no. 2, pp. 199–202, 2021, DOI: 10.13688/j.carol/carroll/nki/CRH.2021.200751.

[10] Y. Chao, W. Yang Junti, and L. G. Xinying, "Analysis on promoting mechanism of hospital informatization construction under improving modern hospital management system," *China health industry*, vol. 16, no. 7, pp. 99–101, 2019.

[11] X. Xie, W. Zhang, H. Wang et al., "Dynamic adaptive residual network for liver CT image segmentation," *Computers and Electrical Engineering*, vol. 91, article 107024, 2021.

[12] A. Khatoon, "A blockchain-based smart contract system for healthcare management," *Electronics*, vol. 9, no. 1, p. 94, 2020.

[13] D. Gu, S. Deng, Q. Zheng, C. Liang, and J. Wu, "Impacts of case-based health knowledge system in hospital management: the mediating role of group effectiveness," *Information & Management*, vol. 56, no. 8, article 103162, 2019.

[14] X. Zhang and Y. Wang, "Retracted article: research on intelligent medical big data system based on Hadoop and blockchain," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, 21 pages, 2021.

[15] H. Luo, J. Liu, C. Li, K. Chen, and M. Zhang, "Ultra-rapid delivery of specialty field hospitals to combat COVID-19: lessons learned from the _Leishenshan_ Hospital project in Wuhan," *Automation in Construction*, vol. 119, article 103345, 2020.

[16] R. Kumar, M. T. J. Ansari, A. Baz, H. Alhakami, A. Agrawal, and R. A. Khan, "A multi-perspective benchmarking framework for estimating usable-security of hospital management system software based on fuzzy logic, ANP and TOPSIS methods," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 1, pp. 240–263, 2021.

[17] X. Luo, C. Zhang, and L. Bai, "A fixed clustering protocol based on random relay strategy for EHWSN," *Digital Communications and Networks*, vol. 9, no. 1, pp. 90–100, 2023.

[18] W. Zhang Gong and L. X. Fei, *Chinese journal of management informatization*, vol. 23, no. 13, pp. 105–107, 2020.

[19] J. Hong and W. Mengying, "Construction and application of hospital intelligent management system," *Chinese Journal of Health Information Management*, vol. 18, no. 2, pp. 164–168, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 3770982, 8 pages
https://doi.org/10.1155/2023/3770982

Research Article

Edge UAV Detection Based on Cyclic Spectral Feature: An Intelligent Scheme

Zhanbin Zhang,[1] **Wenjiang Ouyang**,[2] **Haitao Gao**,[2] **and Xiaojun Jing**[2]

[1]*School of Economics and Management, University of Chinese Academy of Sciences, Beijing, China*
[2]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China*

Correspondence should be addressed to Zhanbin Zhang; zhangzhanbin17@mails.ucas.ac.cn

Received 20 September 2022; Revised 1 November 2022; Accepted 15 November 2022; Published 4 May 2023

Academic Editor: Peiying Zhang

With the commercialization of the fifth-generation mobile communication network (5G), the scale of the unmanned aerial vehicle (UAV) industry has continued to expand. However, the unregistered UAV has caused frequent harassment incidents at international airports, and the problem of UAV crimes is increasing. Radio technology supports long-distance detection of unregistered UAV and can be used as an efficient early warning method for unregistered UAV, which has attracted extensive attention from academia and industry. The classic UAV detection based on remote control signal method faces technical bottlenecks such as being easily affected by environmental noise, high complexity, and low detection accuracy. In the paper, an UAV remote control signal detection method is proposed based on cyclic spectrum features. More specifically, a dataset of UAV remote control signal UAV-CYCset is firstly constructed in the frequency domain. Based on UAV-CYCset dataset, a network architecture is proposed based on improved AlexNet, and the average detection accuracy of the improved model reaches 85% (from -10 dB to 10 dB) according to the simulation experiments.

1. Introduction

In recent years, unmanned aerial vehicle (UAV) has developed rapidly in civilian and has been widely used in aerial photography, agriculture, plant protection, disaster relief, transportation, surveying and mapping, remote sensing, and communications. With the continuous development of UAV, the scale of the industry is also increasing gradually, the fields of application have also been greatly expanded, and the market coverage is increasing year by year. At the same time, with the application of the fifth-generation mobile communication network (5G) in business [1], the data transmission range is wider, the stability is higher, and the delay is smaller. It contributes to the continuous expansion of the application scenarios of UAV and the rapid development of UAV under the impetus [2–4]. With the rapid rise of the UAV industry, many problems have also arisen. Reports of UAV endangering the lives of the public, violating the privacy of others, delaying flights, etc. are not uncommon. As a result, UAV detection has attracted more and more attention in industry and academia.

UAV is widely applied to different fields and is used by the public as one of the means of daily entertainment. It also affects people's normal life to a certain extent and even threatens national security. Most countries have begun to formulate policies to restrict the flight of UAV, the research on related technical means of UAV detection and interference is increased, and the control measures for UAV safety incidents are strengthened.

The main way to monitor and counter UAV is signal detection. The existing detection technologies mainly include (1) radar technology: the radar used in the anti-UAV solution uses one of the following three technologies: pulse (with source), CW (active), and CW (passive) modes. Each method has different characteristics and advantages and disadvantages [5, 6]. (2) Photoelectric technology: use optical cameras to capture scene images, and use infrared imaging or visible light technology to identify targets, to achieve the purpose of tracking and positioning UAV [7, 8]. (3) Sound wave recognition: this technology will store the sound sample data of the UAV in the system in advance, collect the sound data in the environment during the monitoring process, and compare it with the sample data, finally determining the present state of the UAV. In the process of research on UAV flight, some researchers select the feature vector of sound spectrum, including voiceprint energy, MFC.C feature vector, and use support vector machine (SVM) to detect whether the UAV signal exists. In the actual application process, especially for environments with many residents or relatively noisy environments such as games, the method of sound wave identification will be seriously affected, and the detection distance will also be limited. (4) Radio technology: determine whether there is an UAV by detecting whether there is an UAV remote control signal or image transmission signal in the target area [9].

At present, UAV is becoming smaller and lighter, so the radar detection technology is getting more and more difficult to detect civilian UAV. Similarly, optoelectronic technology has high hardware requirements, and an UAV that is tens of meters away may only have a few pixels on the image, making identification difficult. At the same time, the harsh weather environment also causes many difficulties for optoelectronic technology. At present, with the development of technology, the wireless domain signal detection technology has become mature. Based on the electromagnetic signal detection technology combined with the wireless signal detection technology, the research on the technology suitable for detecting the remote-control signal of the UAV is also the current solution to the detection of the cooperation and noncooperation UAV in the complex environment, which is also the core of this paper.

1.1. Related Work. The current UAV detection and identification technology are still based on radar, optoelectronic technology, and wireless signals to detect UAV.

The research and development team of the University of California, San Diego, has built a 5G communication Frequency Modulated Continuous Wave (FMCW) long-distance high-resolution radar system based on 28 GHz phased array in terms of radar detection of UAV signals and detected targets up to 250 meters away with a resolution of 0.15 meters at distance. Ezuma et al. [10] of North Carolina State University in the United States detected and identified UAV through the RF fingerprint of the signal sent by the controller to the micro-UAV and used the energy transient signal to classify the UAV, using the k-nearest neighbor method. For UAV target detection, the average detection accuracy rate is 96.3%. The team from Nanyang Technological University in Singapore [11] designed a low, slow, and small radar target recognition method such as UAV and proposed a two-dimensional regularized complex logarithmic Fourier transform, which better solves the existing signal representation problem. At the same time, the literature proposes a subspace reliability analysis method to optimize the unreliable feature dimension of the conditional covariance matrix. In [12], Yang et al. used spectrum accumulation (SA) and statistical fingerprint analysis (SFA) techniques to estimate the frequency of UAV RF signals and then determine whether there is an UAV in the detection environment. The recognition rate of this method is close to 100% in the range of 2.4 km, and the recognition rate is greater than 90% in the range of 3 km. [13] proposed a radar-assisted positioning method based on 5G millimeter wave, deployed 5G millimeter wave radar, obtained additional features with the help of micro-Doppler characteristics, and then judged and identified the UAV rotor. On this basis, the sine frequency modulation (SFM) parameter optimization method is used to separate multiple UAV and realize the simultaneous detection of multiple UAV under the same conditions. Zhao and Su [14] from the National University of Defense Technology proposed a weak Doppler (m-D) signal evaluation method for UAV based on cyclostationary phase analysis (CPA) and realized the effective detection of small UAV based on radar.

From the mid-to-late 1980s, some related scholars began to explore the cyclostationary characteristics of signals. At first, the first-order statistics and second-order statistics of signals were used to extract signal characteristics [15, 16]. With the continuous evolution of wireless communication technology, Gardner et al. mentioned spectral redundancy and related concepts for the first time, which greatly promoted the research progress related to second-order cyclic statistics. At the end of the 20th century, related scholars successfully applied high-order cycle statistics to practical engineering. For example, the detection and analysis of high-order cycle statistics helped to monitor the failure state of mechanical devices [17]. At present, signal detection methods based on cyclostationary features have attracted extensive attention. Wang et al. proposed a blind detection algorithm for frequency hopping signals based on cyclostationary characteristics, extended the asymptotically optimal χ^2 test method to the detection problem of frequency hopping signals, designed the relevant detection statistics, and completed the Gaussian white noise environment frequency hopping signal detection. The algorithm performs well above -2 dB, the detection probability is 100%, the detection performance drops sharply from -2 dB to -8 dB, and the detection probability is 0% when the signal-to-noise ratio is less than -8 dB [18]. Zhang et al. proposed a neural network spectrum sensing algorithm based on the cyclostationary feature of the signal. By calculating the cyclic autocorrelation function of the signal, perform plane slicing to generate images and label them to generate datasets [19]. The dataset is fed into an artificial neural network with eight hidden layers for training. The experimental results show that the model has good detection performance for the existence of BPSK signal and OFDM signal and still has a detection probability of more than 90% in the case of -20 dB. The algorithm mainly recognizes two kinds of fixed-frequency signals, BPSK and OFDM. Lu et al. proposed an UAV signal identification method based on the contour map of the frequency hopping signal. By performing a short-time Fourier transform on the signal, the contour features of the signal are extracted to construct a three-dimensional matrix and then processed to generate data. The constructed dataset is fed into a convolutional neural network for training [20]. The final model has better recognition accuracy on the dataset, with a detection probability of up to 100% above -10 dB and a detection probability of about 75% at -15 dB. The image acquisition and processing process of the algorithm are relatively complicated, and it is necessary to calculate the maximum value at each time point and then adjust the threshold to obtain the signal contour. After that, the image is processed with contrast and grayscale, so that the entire model consumes a long time.

In practical applications, UAV remote control signals are not only affected by environmental noise but also interfered with by other fixed-frequency communication signals. Therefore, the performance of the proposed UAV signal recognition model based on time-domain image perception is limited by the fixed frequency signal interference problem. The cyclostationary characteristics of wireless signals have the advantage of being insensitive to environmental interference.

1.2. Motivation and Main Contribution. Motivated by mentioned above, to reduce the complexity of the model as much as possible, ensure the recognition performance of the algorithm at low signal-to-noise ratio, and improve the anti-interference ability of the algorithm for fixed-frequency signals during frequency hopping signal detection, this paper proposes a cyclic spectrum-based method. The main contributions are concluded as follows:

(1) A novel UAV-CYCset cyclic spectrum dataset with a signal-to-noise ratio ranging from -10 dB to 10 dB is constructed

(2) A network architecture is proposed based on improved AlexNet

(3) An UAV remote control signal detection method is proposed based on cyclic spectrum features, and the constructed dataset is trained and tested through improved AlexNet

2. System Model

2.1. Cyclostationary Signal. According to the definition, a cyclostationary signal is a nonstationary signal, but it has its own cycle. Usually, there is a lot of information in the cycle when the relevant statistics of the cyclostationary signal change. This paper focuses on its second-order cyclostationary characteristics to illustrate.

Assuming that $x(t)$ is a nonstationary signal and has zero mean, it becomes $x(t)x^*(t-\tau)$ after secondary transformation. The following is the time-varying correlation function expression for $x(t)$:

$$R_x(t,\tau) = E\{x(t)x^*(t-\tau)\}. \tag{1}$$

Assuming that the period of $x(t)x^*(t-\tau)$ is T_0, then taking the relevant theory of Fourier series as a reference, the sample collection of $x(t)x^*(t-\tau)$ with the period of T_0 is performed, so the following can be obtained. Statement expression:

$$\begin{aligned} R_x^\infty(t,\tau) &= E\{x(t)x^*(t-\tau)\} \\ &= \lim \frac{1}{2N+1}\sum_{n=-N}^{N} x(t+nT_0)x^*(t+nT_0-\tau). \end{aligned} \tag{2}$$

Since $R_x(t,\tau)$ takes T_0 as the period, the relevant function can be expanded through the Fourier series, and the function expansion is as follows:

$$R_x^\infty = \sum_{m=-\infty}^{\infty} R_x^\infty(t,\tau)e^{-j2\pi\alpha t}. \tag{3}$$

In the above formula, $\alpha = m/T_0$, and its Fourier series is

$$R_x^\infty(\tau) = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} R_x(t\,;\tau)e^{-j2\pi\alpha t}dt. \tag{4}$$

Bring equation (2) into equation (4) to get:

$$R_x^\infty(\tau) = \lim_{T\longrightarrow\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t)x^*(t-\tau)e^{-j2\pi\alpha t}dt. \tag{5}$$

The above formula expresses the time average of the correlation function. The coefficient $R_x^\infty(\tau)$ represents the degree of cyclic autocorrelation of the signal at frequency α, also known as the cyclic autocorrelation function. The value that obtains a nonzero value of $R_x^\infty(\tau)$ is called the cyclic frequency of the signal. This parameter mainly reflects the cyclostationary characteristics of the signal. Usually, there may be different cyclic frequencies in the corresponding specific signal.

If $\alpha = 0$, we have

$$R_x^0(\tau) = x\left(t+\frac{\tau}{2}\right)x^*\left(t-\frac{\tau}{2}\right)_t. \tag{6}$$

At this time, after $R_x^\infty(\tau)$ is degraded, it becomes the autocorrelation function of the stationary signal. If the signal satisfies $R_x^\infty(\tau) = 0, \forall \propto \neq 0$, it can be regarded as a stationary signal; and if it satisfies $R_x^\infty(\tau) \neq 0, \exists \propto \neq 0$, it belongs to a cyclostationary signal.

Cyclostationary signal itself does not have statistical characteristics. After a series of calculations, it can be found that some of its mathematical characteristics are periodic. The mathematical expectation in the first-order statistics of the cyclostationary signal and the signal autocorrelation function in the second-order statistics have been verified to be periodic. Therefore, it is possible to find and select an appropriate way to process the signal and convert the signal characteristics, which is beneficial to find the characteristics of the signal in essence. And whether it is first-order or second-order cyclostationary, its characteristics are related to the frequency shift signal, which mainly depends on the way of operation.

Because the cyclostationary signal processes the signal features in a corresponding way and completes the feature extraction, the processing of such signals is obviously different from the traditional signal processing methods. First of all, when processing cyclostationary signals, the signal characteristics obtained by transformation are the main components. Compared with directly processing the original signal, the processing of the cyclostationary signal will extract the simplified signal characteristics of the original signal, which will obviously reduce the complexity. Second, when analyzing the cyclostationary signal, the target is statistical information, which can reduce the interference caused by the noise with stationary characteristics and improve the antinoise ability. Third, when dealing with actual signals, choosing cyclostationary signals for processing is more in line with reality, which not only ensures the rationality of the results but also makes the processing process easier. It is also because of the characteristics of the cyclostationary signal itself that the use of cyclostationary detection to receive signals has become the most widely used method. Cyclostationary detection is mainly used in the signal processing process to judge whether the signal has cyclostationary characteristics; second, if it is known that the signal has cyclostationary characteristics, even on the premise that the signal has cyclostationary frequency characteristics, this method is usually used to judge whether there is a known signal in the signal. The above applications make the cyclostationarity detection method more and more common in the current signal processing process.

2.2. Cycle Spectrum. The realization of the cyclic spectrum is mainly based on the digital fast Fourier transform. Common cyclic spectrum implementation algorithms mainly include three categories. The first is the time domain smooth estimation algorithm, and the formula is as follows:

$$S_x^{\infty}(t,f) = \frac{1}{\mathrm{KM}} \sum_{u=0}^{\mathrm{KM}-1} \Delta f X_{1/\Delta f}\left(t - \frac{u}{k\Delta f}, f + \frac{\alpha}{2}\right) X_{1/\Delta f} \cdot \left(t - \frac{u}{k\Delta f}, f + \frac{\alpha}{2}\right), \tag{7}$$

In the above formula, $X_{1/\Delta f}(t,f)$ represents the smooth DFT transform output; $\Delta f = 1/(N-1)T_S$ represents the spectral resolution. $1/\Delta f$ represents the length of the segment.

The second is the frequency domain smoothing estimation method, and the formula is as follows:

$$S_x^{\propto}(t,f)_{\Delta f} = \frac{1}{M} \sum_{v=-M-1/2}^{M-1/2} \frac{1}{\Delta t} X_{\Delta t}\left(t, f + \frac{\alpha}{2} + vF_S\right) X_{\Delta t}^{*} \cdot \left(t, f - \frac{\alpha}{2} + vF_S\right), \tag{8}$$

where $X_{\Delta t}(t,f)$ represents the output after sliding DFT operation. T_S represents the time sampling increment; $\Delta f = MF_S$ represents the spectral smoothing gap width; the total number of samples in the data segment of the time interval Δt is $N = (\Delta t/T_S) + 1$.

To sum up, the process that the output obtained after demodulation is processed by conjugate multiplication is the operation process of the FFT accumulation algorithm, so the computing power of the computer is relatively high when performing the two-dimensional fast Fourier transform operation, which usually requires more memory and takes more time to complete. The instantaneous correlation function algorithm first requires the calculation of the autocorrelation function of the nonstationary signal, then completes the transformation in the time domain and frequency domain, and then estimates the cyclic spectrum.

2.3. The Generation of Dataset. The frequency hopping signal used in this experiment is also generated by the modeling tool in MATLAB. The center frequency of the frequency hopping signal is 2.4 GHz, the signal rate is 50 Kb/s, the number of bits per hop is 50, and the bandwidth of the frequency hopping signal is 9.8 MHz. After the frequency hopping signal is generated, it also needs to go through the Rayleigh channel model containing Gaussian white noise. In this experiment, after obtaining the noisy signal noise, it is necessary to calculate the cyclic autocorrelation function of the signal and then perform the FFT operation on the function to obtain the spectral correlation function of the signal. Based on the theoretical knowledge described in the previous chapter, the spectral correlation function obtained using correlation operation has noise, signal spectral correlation function, and cross-spectral correlation function. The cyclic spectrum is to expand the spectral correlation function of the obtained noisy signal on the α-axis and the f-axis, so that the cyclic spectrum can be obtained, where α is an integer multiple of the fundamental frequency of the signal, and f represents the frequency of the signal. The cyclic spectrum is a three-dimensional image, and the height of its vertical axis is normalized to determine whether there is a main signal. Since the image we put into the convolutional neural network is two-dimensional, the experiment needs to map the three-dimensional image on the two-dimensional plane to ensure that the characteristics of the cyclic spectrum can be preserved to the greatest extent in the two-dimensional plane. In this experiment, the $\alpha - f$ plane is used as the benchmark to map the vertical axis of the plane. The experiment uses color to represent the amplitude of the spectral function at this point, as shown in Figure 1 below. The darker the color, the higher the amplitude value. Shallower values represent lower amplitude values.

The computational complexity of the cyclic spectrum is high. In this experiment, 2048 sample points are intercepted from the received signal to calculate the total to obtain the cyclic spectrum of the signal. Since the convolutional neural network needs control samples, this experiment uses the same algorithm to generate the cyclic spectrum of noise when generating the cyclic spectrum of the noisy signal, as shown in Figure 2. The signal-to-noise ratio range of the frequency hopping signal collected in this experiment is also -10~10 dB, with 1 dB as an interval. 100 signal time domain images and noise time-domain images were collected under different signal-to-noise ratio conditions.

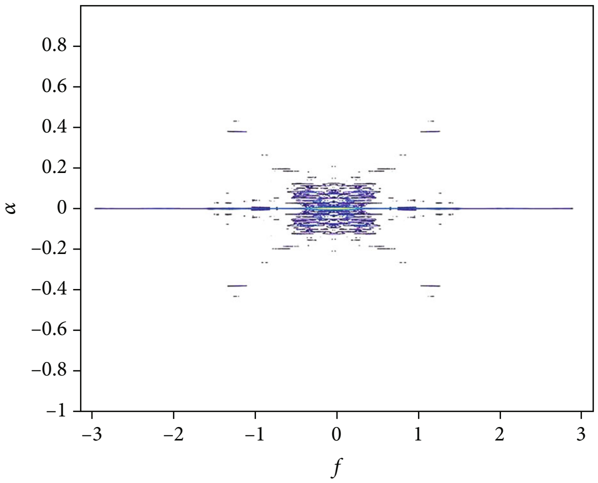


Figure 1: Frequency hopping signal cycle spectrum.

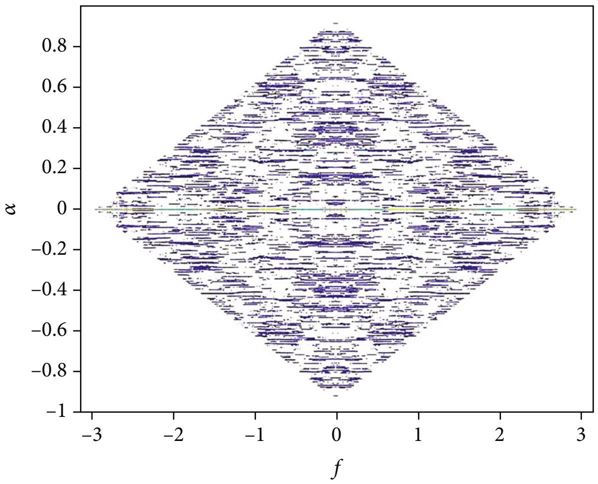


Figure 2: Noise cyclic spectrum.

After the acquisition of the image is completed, the pixels where the irrelevant information such as the horizontal and vertical coordinates in the input image are located are deleted, and only the cyclic spectrum image is retained. Then resize the image to 227*227*3, where 227 is the length and width of the image, and 3 is the number of channels of the image. After the image size processing is completed, the signal image and the noise image are labeled, respectively, and stored as record files. Each file includes the corresponding image and its corresponding label. The signal time domain image label is 1, and the noise image label is 0. Finally, generate the UAV-CYCset cyclic spectral dataset of UAV signals. In the experiment, 4200 time-domain images were obtained through MATLAB simulation, of which there were 100 signal and noise images under each signal-to-noise ratio. In this experiment, 70 signal images and 70 noise images under each signal-to-noise ratio are taken from UAV-CYCset, a total of 2960 images are used as the training set, and the remaining 1240 images are used as the test set.

3. Proposed Schemes

In the pretraining stage, this experiment considers a traditional convolutional neural network (CNN) for training. After the model converges, the expected effect is not achieved, and the recognition rate of the training set is lower than 85%. Considering that the cyclic spectrum of the signal has more feature points than the time domain image, and the size of the generated image is larger, the CNN network may not meet the experimental requirements of this experiment. After investigation and multiple pretraining experiments, the AlexNet model was selected as the basic experimental model in this experiment.

3.1. Model Architecture and Improvements. Under the premise of ensuring the recognition performance of the model, AlexNet [21] has fewer convolution layers, which reduces the complexity and time consumption of the model, and the input is 228*228, which reduces the impact on the recognition performance caused by the clipping and scaling of the circular spectrum image. The model has a total of eight hidden layers, of which the first five are convolutional layers, and the rest are fully connected layers. Among all convolutional layers, only layers 1, 2, and 5 use pooling operations. The improved AlexNet architecture is shown in Figure 3, and the parameters of AlexNet network are listed in Table 1.

The input data of the first layer is the image with the original size of 227*227*3, where 227 represents the image size, 3 represents the number of channels of the image, the convolution kernel size of this layer is 11*11, and the number of channels is the same as the number of image channels. The pixel layer output by the first layer is used as the input data of the second layer, the size of the pixel layer is 27*27*48, the pixel layer output by the second layer is used as the input data of the third layer, and the size of the pixel layer is 13*13*128. The pixel layer output by the third layer is used as the input data of the fourth layer. The size of the pixel layer is 13*13*192.

The fifth layer is the same as the fourth layer, and the output after the convolution operation is still 13*13*192. The sixth layer is a fully connected layer, which inputs data with a size of 6*6*256 and convolves the input data through filters of the same size. The 4096 data output in the seventh layer is fully connected to the 4096 neurons in this layer, and then the 4096 data formed by the activation function ReLU and Dropout operation are processed. The 4096 data input in the 8th layer is fully connected to the 1000 neurons in this layer, and the trained values are outputted externally after training.

This experiment has been optimized based on AlexNet. The ReLU function is used as the activation function in the AlexNet network model. The ReLU function can accelerate the convergence and solve the problem of gradient disappearance. However, because the negative semiaxis of the ReLU function is 0, the weight may not be updated because the derivative is 0. Therefore, in this experiment, the Swish activation function is used to replace the ReLU function. The Swish function formula is as follows:

$$f(x) = x\frac{1}{1+e^{-x}}. \tag{9}$$

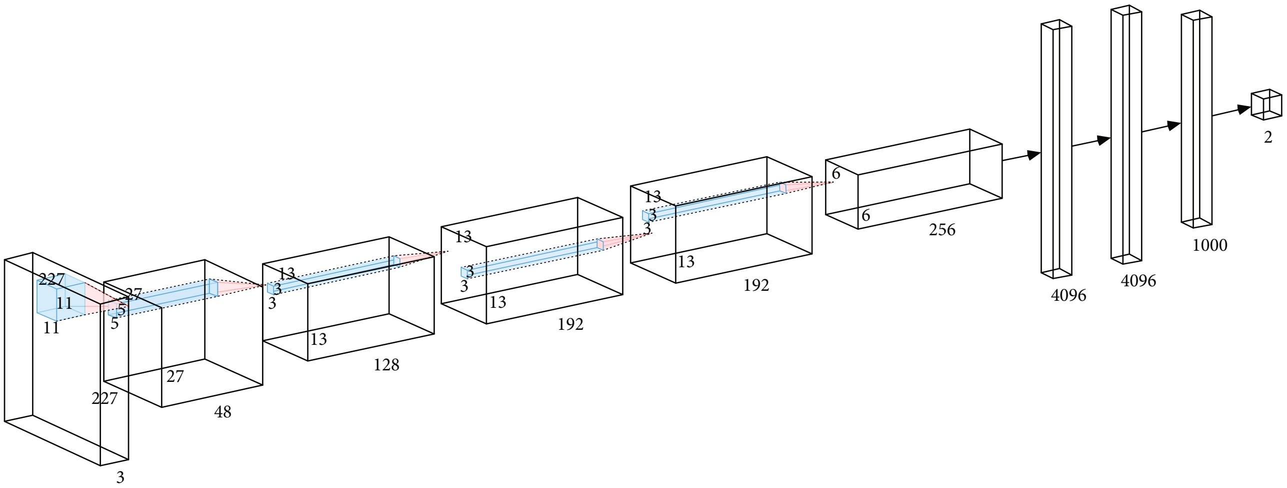


Figure 3: The architecture of the proposed improved AlexNet [21].

Table 1: The parameters of AlexNet network.

Model structure	Model parameter
Convolution layer1	48 (11 × 11)
MaxPool layer1	3 × 3
Convolution layer2	128 (5 × 5)
MaxPool layer2	3 × 3
Convolution layer3	192 (3 × 3)
Convolution layer4	192 (3 × 3)
Convolution layer5	256 (3 × 3)
MaxPool layer3	3 × 3
Fully connected layer1	4096
Fully connected layer2	4096
Fully connected layer3	1000
Fully connected layer4	2

Compared with the ReLU function, the Swish function does not have a derivative of 0. After pretraining and the comparison and verification of the ReLU function, the Swish function can better solve the problem that the weights caused by the ReLU function cannot be updated during the training process. At the same time, the Swish function also has a certain improvement in model overfitting. Since Adam requires fewer computing resources than RMSProp, this experiment chooses to use the Adam algorithm as the optimizer in model training.

3.2. Model Training. In the process of model training in this experiment, the signal sample images and noise sample images in the training set are scrambled, respectively, to generate sample queues. In each training, 50 samples are drawn from the team leaders in the two queues to form a small training set with a size of 100, and the small training set is sent to the neural network for training. After each training, the signal samples and noise samples in the small training set are put into the tail of their corresponding sample queues, respectively, and the previous operations are repeated to train the convolutional neural network.

In this experiment, a 10-fold crossover method is also used to test the accuracy of the algorithm. At the same time, 7 copies are selected as training data, and the rest are used as test data. The experiments are completed in turn, and the corresponding accuracy rate is obtained in each round. Take the average after 10 training runs, and the result is an estimate of the accuracy of the algorithm.

4. Simulation and Discussion

Since this experiment uses the self-built data set UAV-CYCset for training and testing, and there is no publicly related data set in the network, this experiment mainly compares the recognition performance of traditional algorithms. The comparison algorithm selected in this experiment is the blind detection algorithm of frequency hopping signal based on cyclic autocorrelation proposed by Fan Haining. The basic principle of the algorithm is to assume that the received signal is $x(n)$, the received signal contains the frequency hopping signal and noise, and the cyclic autocorrelation function is calculated for the received signal $x(n)$. The formula is as follows:

$$\widehat{R}_{xx}^{\alpha}(m) = \frac{1}{L}\sum_{n=0}^{L-1} x(n+m)x^{*}(n)\exp\left(-j2\pi\alpha n\right). \qquad (10)$$

In the formula, L is the length of the received signal. After that, calculate the average power $\bar{P}_x$ of the received signal with the following formula:

$$\bar{P}_x = \frac{1}{L}\sum_{n=0}^{L-1} |x(n)|^2. \qquad (11)$$

After that, a Gaussian white noise sequence $v(n)$ with the same length as the received signal is generated, the power of the Gaussian white noise is required to be equal to the

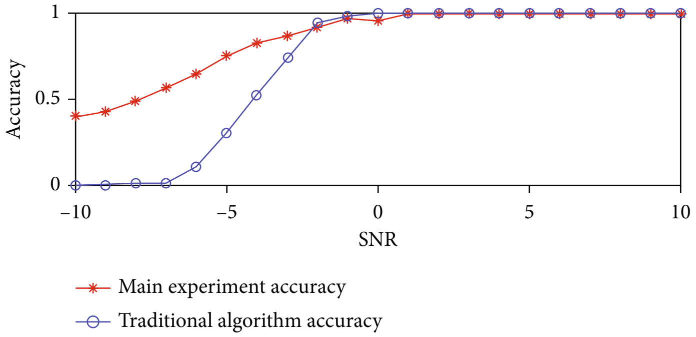


Figure 4: The comparison of accuracy between AlexNet and benchmark.

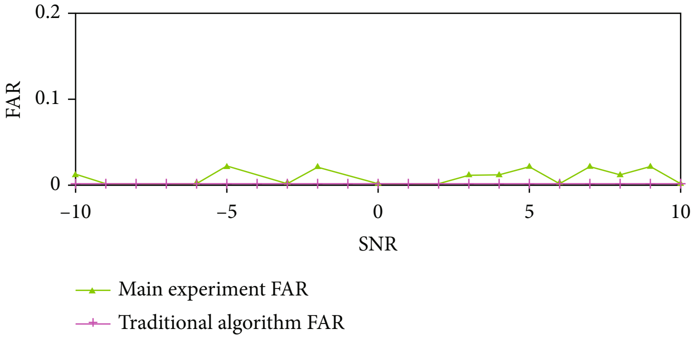


Figure 5: The comparison of false alarm rate between AlexNet and benchmark.

average power of the received signal, and then the cyclic autocorrelation function of $v(n)$ is calculated to estimate $\widehat{R}_w^{\alpha}(m)$. Select the maximum value of $\widehat{R}_w^{\alpha}(m)$ of $\max\limits_{\alpha}\widehat{R}_w^{\alpha}(m)$ as the decision threshold th, artificially add a correction weight, set it as ω, and then whether the frequency hopping signal exists, the judging method is that if there is at least one α such that $|\widehat{R}_w^{\alpha}(m)| > th * \omega$, it is judged that there is a frequency hopping signal; otherwise, it is judged that there is only noise. The data length of the signal in this experiment is $L = 2048$, and the correction weight is 1.35.

In this experiment, the signal sample is a positive example. After the correlation operation, the recall rate of the convolutional neural network model can reach 85.05%, and the accuracy rate can reach 99.112%. It can be seen from the data that the convolutional neural network has a good recognition performance for the cyclic spectrogram of the UAV frequency hopping signal. The recall rate obtained by the comparison algorithm through experiments is 69.69%, and the precision rate is 100%. Since the sample contains signal and noise samples with different signal-to-noise ratios from -10 dB to 10 dB, the corresponding experiment results are exhibited as it is shown in Figures 4 and 5.

It can be seen from Figure 4 that the accuracy of the convolutional neural network in identifying signal samples has an upward trend. Although the recall rate of the convolutional neural network in the entire test set can reach 85.05%, the recognition rate of signal samples with low signal-to-noise ratio is poor. When the signal-to-noise ratio is lower than -5 dB, the probability of identifying the signal is less than 80%. If when the signal-to-noise ratio is -10 dB, the probability of identifying the signal is only 41%. Although the convolutional neural network model shows a downward trend as the signal-to-noise ratio decreases, this does not affect the excellent recognition performance of the neural network model in the case of high signal-to-noise ratio. If the signal-to-noise ratio exceeds 1 dB, the probability of identifying the signal can reach 100%. The counterexample sample in the experiment is the cyclic spectrogram of noise. As shown in Figure 5, the green line in the figure represents the false alarm rate of the main experiment, that is, the recognition rate of identifying the noise sample as a signal. The neural network performs well in the entire range of signal-to-noise ratio, the overall false alarm rate is only about 0.77%, and the signal-to-noise ratio change has no effect. The traditional algorithm has good recognition performance when the signal-to-noise ratio is greater than 0 dB, the signal recognition probability drops sharply between -2 dB and -7 dB, and the recognition rate is basically 0 when the signal-to-noise ratio is less than -7 dB.

Compared with the traditional algorithm, the model generated in this experiment has little difference in the case of high signal-to-noise ratio and performs well in the false alarm rate. As the signal-to-noise ratio decreases, the recognition rate of the network model will not drop drastically like traditional algorithms. At the same time, when the UAV is receiving the frequency hopping signal, it is possible to receive not only environmental noise but also signals emitted by other transmitters. However, since most signals do not have cyclic characteristics, the cyclic spectrum is used as a convolutional neural network. The input of the model ensures the anti-interference performance of the neural network under the condition of other fixed-frequency signals to a certain extent and has great application value.

5. Conclusion

In this paper, a detection method of UAV remote control signal based on cyclic spectrum feature is proposed. First, the basic theory of the current cyclostationary theory is introduced, and the basic characteristics of the cyclostationary signal and the cyclic spectrum can be used as the theoretical basis for the detection of the frequency hopping signal. After that, the establishment of the cyclic spectral sample dataset is conducted. Based on the UAV-CYCset dataset of UAV remote control signal frequency domain, a network architecture is proposed based on improved AlexNet, and the average detection accuracy of the improved model reaches 85% (-10 dB-10 dB), which demonstrates the feasibility of using cyclic spectrogram as input to detect UAV frequency hopping signal using convolutional neural network.

Data Availability

The dataset is available. If need, contact the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Mu, X. Jing, Y. Zhang, Y. Gong, R. Zhang, and F. Zhang, "Machine learning-based 5G RAN slicing for broadcasting services," *IEEE Transactions on Broadcasting*, vol. 68, no. 2, pp. 295–304, 2022.

[2] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: applications, challenges, and open problems," *IEEE Communications. Survey*, vol. 21, no. 3, pp. 2334–2360, 2019.

[3] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communication Surveys and Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[4] J. Mu, F. Zhang, Y. Cui, J. Zhu, and X. Jing, "Non-cooperative UAV detection with adaptive sampling of remote signal," in *International Wireless Communications and Mobile Computing (IWCMC)*, pp. 791–795, Harbin City, China, 2021.

[5] N. Zhao, F. Cheng, F. R. Yu et al., "Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Transactions on Communications*, vol. 66, no. 5, pp. 2281–2294, 2018.

[6] C. Wang, J. Tian, J. Cao, and X. Wang, "Deep learning-based UAV detection in pulse-Doppler radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[7] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 879–892, 2017.

[8] Y. Bazi and F. Melgani, "Convolutional SVM networks for object detection in UAV imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3107–3118, 2018.

[9] X. Ai, L. Zhang, Y. Zheng, and F. Zhao, "Passive detection experiment of UAV based on 5G new radio signal," in *2021 Photonics & Electromagnetics Research Symposium (PIERS)*, pp. 2124–2129, Hangzhou, China, 2021.

[10] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-UAV detection and classification from RF fingerprints using machine learning techniques," in *2019 IEEE Aerospace Conference*, Big Sky, MT, USA, 2019.

[11] J. Ren and X. Jiang, "Regularized 2-D complex-log spectral analysis and subspace reliability analysis of micro-Doppler signature for UAV detection," *Pattern Recognition*, vol. 69, pp. 225–237, 2017.

[12] S. Yang, H. Qin, X. Liang, and T. Gulliver, "An improved unauthorized unmanned aerial vehicle detection algorithm using radiofrequency-based statistical fingerprint analysis," *Sensors*, vol. 19, no. 2, p. 274, 2019.

[13] J. Zhao, X. Fu, Z. Yang, and F. Xu, "Radar-assisted UAV detection and identification based on 5G in the internet of things," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 2850263, 12 pages, 2019.

[14] Y. Zhao and Y. Su, "Cyclostationary phase analysis on micro-Doppler parameters for radar-based small UAVs detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2048–2057, 2018.

[15] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 416–429, 2000.

[16] A. Abdelmutalab, K. Assaleh, and M. El-Tarhuni, "Automatic modulation classification based on high order cumulants and hierarchical polynomial classifiers," *Physical Communication*, vol. 21, pp. 10–18, 2016.

[17] L. Izzo and A. Napolitano, "Multirate processing of time series exhibiting higher order cyclostationarity," *IEEE Transactions on Signal Processing*, vol. 46, no. 2, pp. 429–439, 1998.

[18] M. Oner, "Spectral correlation of a digital pulse stream modulated by a cyclostationary sequence in the presence of timing jitter," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 339–342, 2009.

[19] L. Izzo, L. Paura, and G. Poggi, "An interference-tolerant algorithm for localization of cyclostationary-signal sources," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1682–1686, 1992.

[20] Y. Xie, P. Jiang, Y. Gu, and X. Xiao, "Dual-source detection and identification system based on UAV radio frequency signal," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[21] T. Technicolor, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9816845, 1 page
https://doi.org/10.1155/2023/9816845

Retraction

Retracted: Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning

Wireless Communications and Mobile Computing

Received 8 August 2023; Accepted 8 August 2023; Published 9 August 2023

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) Peer-review manipulation

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] M. Li, W. Sun, Q. Jia et al., "Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1931965, 7 pages, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1931965, 7 pages
https://doi.org/10.1155/2022/1931965

Research Article

Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning

Mo Li,[1] WeiSheng Sun,[1] Qiaoran Jia,[2] Yilin Cui,[1] Shiru Li,[2] Liping Leah Wu,[2] Lixian Zheng,[1] and Guanghui Qiao[3]

[1] *City University of Macau School of Business Macau Special Administrative Region, 999078, China*
[2] *School of International Tourism and Management, City University of Macau Macau Special Administrative Region, 999078, China*
[3] *College of Tourism and Urban-Rural Planning, Zhejiang University of Technology and Industry, Hangzhou, Zhejiang, China 310000*

Correspondence should be addressed to Lixian Zheng; 192911083@st.usst.edu.cn

Received 24 August 2022; Revised 14 September 2022; Accepted 20 September 2022; Published 15 October 2022

Academic Editor: Peiying Zhang

Aiming at the defect that the click-through rate of marketing advertisements cannot provide accurate prediction results for the company in time in the marketing strategy of Internet companies, this paper uses a deep learning algorithm to establish a prediction model for the click-through rate of marketing advertisements. The suggested model is called high-order cross-feature network (HCN). Furthermore, this paper also introduces the combination of feature vectors into the graph structure and as the nodes in the graph; therefore, the graph neural network (GNN) is used to obtain the high-level representation ability of structured data more fully. Through numerical simulations, we observed that HCN has the capability to provide Internet companies with more accurate advertising business information, user information, and advertising content. Moreover, HCN model is more reasonable to adjust the advertising strategy and can provide better user experience. The simulation outcomes indicate that the suggested HCN approach has noble adaptability and high correctness in forecasting the click-through rate of marketing advertisements. We observed that this improvement, in terms of predictions precisions and accuracies, can be as high as 17.52% higher than the deep neural network (DNN) method and 10.45% higher than the factorization network (FM) approach.

1. Introduction

In traditional applications, the historical click-through rate of advertisements is usually counted as a prediction result, and then the advertisements are sorted according to the historical click-through rate. However, the problem with this method is that the statistical click-through rate is only a fixed value, and it is difficult to carry out personalized advertising according to the preferences of specific users [1]. In fact, the difference between advertising space, users and advertising content will affect the probability of users clicking on the advertisement. In recent years, the development of machine learning algorithms has led researchers to consider the problem of advertising click-through rate estimation as a typical two-class problem, that is, to directly predict the probability that an upcoming advertisement will be clicked by the current user.

Aoying et al. [2] proposed a new framework to train logistic regression models in parallel, and experiments showed that using this framework can reduce the training time by an order of magnitude. Although traditional linear models have application advantages, these models lack the ability to learn cross-features, and feature engineering of data often directly affects the performance of the model. Therefore, in order to improve the effect of the model, it is necessary to spend a lot of time artificially constructing combined features for better performance. However, even experienced data experts cannot construct all the hidden feature information. Therefore, the authors in [3] proposed to learn the original features through GBDT, and then add the number of the leaf nodes of each tree as new features to the logistic regression model. The suggested approach has the capability to solve the feature combination problem of linear models. Of course, in real scenarios, the CTR estimation is not a simple linear problem. Therefore, some researchers have begun to consider how to increase the nonlinear relationship to solve the CTR estimation problem. Agarwal et al. [4] used the hybrid logistic regression algorithm to directly introduce piecewise linearity in the original data space to fit the high-dimensional nonlinear data distribution. And it has end-to-end learning ability, which saves a lot of artificial feature design.

Scholars believe that different keywords have different probability of being clicked by users. Therefore, by analogy to advertisements, the distribution of click-through rate also follows this principle. Therefore, the clustering algorithm is used to cluster keywords to obtain new information, so as to improve the accuracy of the model when the user interaction data is insufficient. Gai et al. [5] used the relationship between search term and advertisement pair to build a user click association graph and extracted the statistical information related to the search term or advertisement from the perspective of the relation graph. However, it is very inappropriate to consider all feature intersections, and not all feature combinations will bring benefits, so they use decision trees to train the model and use the feature importance output of the decision tree to selectively prune features to remove redundancy. References [6–8] improve the prediction performance of the model. Online advertising display is a very promising field. How to display the right advertisement to the customer at the right time and place is a core algorithm problem of the advertisement display system. Therefore, accurate prediction of the click-through rate of advertisements has become one of the most important technologies in advertising algorithms [9, 10]. The following are the major contributions of our research:

(i) We use a deep learning algorithm to establish a prediction model for the click-through rate of marketing advertisements

(ii) The suggested model HCN also introduces the combination of feature vectors into the graph structure, and the graph neural network (GNN) is used to obtain the high-level representation ability of structured data more fully

(iii) The simulation outcomes indicate that the suggested HCN approach has noble adaptability and high correctness in forecasting the click-through rate of marketing advertisements

The rest of the paper is structured as follows: the higher-order cross-feature depth model (HCN) is deliberated in the next Section 2. In Section 3, dataset and training of deep models are discussed in more detail. Numerical experiments, validations, and obtained results are discussed in Section 4. Finally, we summarize the paper in Section 5 and discuss some future insights that can be used by the researcher to take our work into the next levels.

2. Higher-Order Cross-Feature Depth Model

2.1. High-Order Cross-Feature Network. For a sample, since each domain generally has one and only sparse feature value of 1, after the embedding layer index is taken, a domain has only one embedding vector $v_{1,i}$, and i is the domain index. Then, the cross-feature vector of the first layer is mathematically expressed.

$$Z_1^{i,j} = NDP\left(v_{1,i} \cdot v_{1,j}\right) w_1^{i,j}. \tag{1}$$

In Equation (1), $Z_1^{i,j}$ is the cross eigenvector of the eigenvectors $v_{1,i}$ and $v_{1,j}$ of domain i and domain j in the first layer. Similarly, NDP is the dot product normalization operation, and $w_1^{i,j}$ represents the vector-level weight for domain combination.

For the obtained cross-feature vector, the vector representation of the new sparse feature can be reconstituted according to the domain belonging, and then the calculation formula of each domain feature vector of the l^{th} layer is expressed using the following:

$$v_{l,t} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} |t \in \{i, j\}| Z_{l-1}^{i,j} \cdot h_1^{i,j} + v_{1,t}. \tag{2}$$

In Equation (2), t represents the index of the domain, n is the number of domains, and $v_{l,t}$ represents the newly reconstituted sparse feature vector representation in the domain numbered t in the l^{th} layer. This should be noted that $t \in \{i, j\}$ means that if t is equal to i or j, then the value is 1, otherwise the value is 0. Furthermore, $h_1^{i,j}$ is the weight vector to be learned. The vector $v_{l,t}$ reweights and sums all the cross-feature vectors related to the domain t in the previous layer to reconstruct the new sparse feature vector representation and introduces the embedding vector $v_{l,t}$ to avoid the loss of the underlying information. In the next step, we recalculate the cross eigenvectors on the newly obtained vector representation, then the cross-feature vector of the first layer, as stated in equation, is recomputed using the following:

$$Z_l^{i,j} = NDP\left(v_{l,i} \cdot v_{l,j}\right) w_l^{i,j}. \tag{3}$$

Similarly, next we iteratively calculate the above three formulas, i.e., Equations (1)–(3), in order to obtain the cross-feature information of different orders. In order to preserve the cross information of different levels as much as possible, the HCN sums the cross-feature vectors of each layer and outputs as

$$X_l^{\text{out}} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{r=1}^{k} z_l^{i,j,r}, \tag{4}$$

where k is the dimension size of the embedding vector. If the Sigmoid activation function is used after summing the X_l^{out} of all layers, then it can be used as a separate high-order cross-feature model HCN, and the structure is shown in Figure 1. If the linear module and the depth module are combined, then the calculation formula of the DCFM model is illustrated as

$$\widehat{y}_{\mathrm{DCFM}} = \sigma\left(w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{l=1}^{L} X_l^{\mathrm{out}} + \mathrm{MLP}(v_{\mathrm{concat}})\right), \quad (5)$$

where MLP is the output of the depth module, and L is the number of HCN layers. The HCN model continuously reconstructs new feature vector representations for sparse features, and then uses crossover operations to form higher-order cross-feature information. The entire process of the HCN model is shown in Figure 1.

2.2. The Choice of Learning Rate Method. In the advertisement click-through rate estimation scenario, the data set is very huge, often millions or even tens of millions. If the traditional fixed learning rate method is used, training will be very time-consuming. At the same time, if the learning rate is set too large, the training time will be shortened, but it may oscillate around the minimum value and fail to converge, and the accuracy will be greatly reduced. If the learning rate is set too small, although the accuracy will be improved, it will be very time-consuming. In order to find a perfect balance between training time and accuracy, this article will use a degenerate learning rate, also known as learning rate decay. The degenerate learning rate combines the advantages of two methods of large learning rate and small learning rate in the training process, that is, using a large learning rate at the beginning of training to speed up training. After training to a certain extent, reduce the learning rate to improve the accuracy and reach the convergence state. The calculation expression is mathematically expressed and given in the following:

$$\mathrm{learning\_rate} = \mathrm{learning\_rate\_base} * \mathrm{learning\_rate\_decay} * \frac{\mathrm{global\_step}}{\mathrm{decay\_step}}. \quad (6)$$

In Equation (6), learning_rate_base represents the initial value of the learning rate. learning_rate_decay represents the learning rate decay coefficient, and global_step represents the number of rounds of sample training. Decay_step represents the step of decay. For example, set learning_rate_base to 0.1, learning_rate_decay to 0.9, global_step to 50, and decay_step to 10. Then the learning_rate is 0.1 at the beginning of the training, the learning_rate is 0.09 when the training reaches the 20th round, and so on, and the learning_rate is 0.081 at the 40th round, as made known in Table 1.

This can be easily understood and comprehended from the values given in Table 1 that the learning rate changes relatively large at the beginning, and the later changes are small, which can meet the needs of a large learning rate in the early stage to speed up the training speed, and a small learning rate in the later stage to improve the accuracy and achieve the purpose of convergence.

2.3. Selection of Activation Function. In order to avoid gradient dispersion and speed up the training speed, the ReLU activation function is generally used when selecting the activation function. The function image of the ReLU activation is shown in Figure 2.

It can be seen from Figure 2 that the ReLU activation function outputs all values less than 0 as 0. The derivative of some functions greater than 0 is a fixed value, so when backpropagating, the calculation is simple, and the training speed can be accelerated. However, since all values less than 0 are replaced with 0, therefore there is a great possibility that all data will be 0, and the model cannot continue during the backpropagation process. In order to speed up the training speed without discarding all negative values in the ReLU, this article will use the activation function which is known as the LeakyReLU and is a variant of the classical ReLU activation. The image of the LeakyReLU function is shown in Figure 3.

3. Training of Deep Models

3.1. Dataset Information. This paper tests the model effect on three general public datasets. The specific data set statistics are shown in Table 2. The Criteo dataset is a dataset released by Criteo, a world-renowned advertising company, during a display advertising competition. The dataset comes from the logs recorded by users when they visit web pages and contains 40 million data samples, each of which has 13 numerical features and 26 discrete features. All features are anonymous, so there is no way to know the specific meaning of the features. The Conversion dataset contains historical conversion feedback for over 1 million clicked ads over a 2-month period. Each row of samples represents a historically clicked display advertisement information, which consists of 8 numerical features and 9 discrete features. The discrete features are hashed into 32-bit integer numbers for the purpose of protecting user privacy, and some features may have missing values. To reduce the size of the original data, the dataset has been down sampled. Similarly, the Avazu dataset is a dataset for real-time advertising algorithm competition shared by a foreign advertising company known as Avazu in 2015. In fact, the dataset records information such as the time, advertisement position, URL, and device type when an advertisement is displayed within 10 days. All features of the sample are discrete features and have been desensitized.

3.2. Model Parameter Settings. For dataset processing, refer to the process of processing Criteo datasets in the open source framework Paddle. For discrete features, after counting the number of occurrences in the entire data set, the features with less than a certain number of occurrences are attributed to one feature; for numerical features, the features with more than 95% quantile values are directly assigned to 95% after ascending order. The corresponding value is located at the quantile of the dataset. Each feature is then mapped to an integer index value using a label encoding algorithm. In the model Embedding layer, the embedding vector corresponding to the sparse feature is found directly

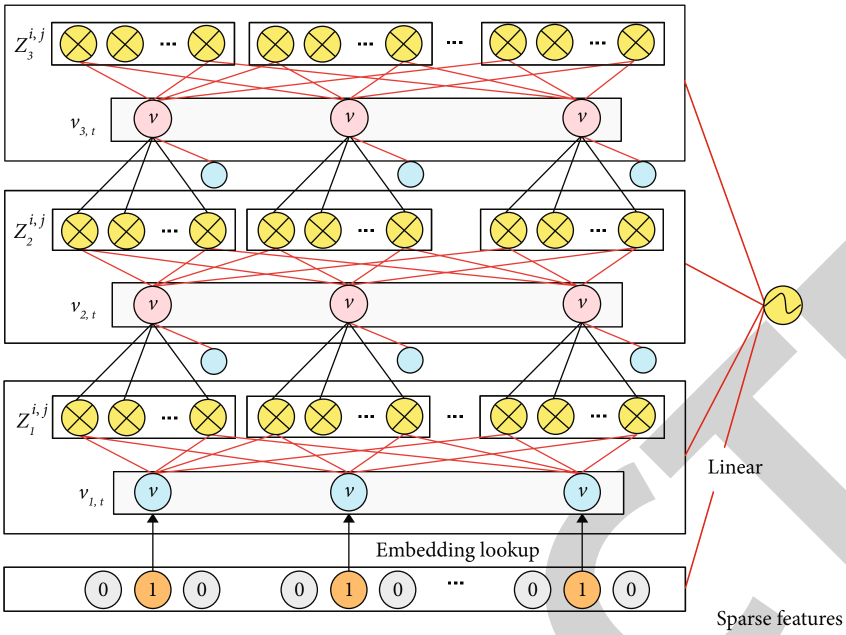


Figure 1: The three-layer HCN model structure.

Table 1: The learning rate attenuation.

Number of training rounds	Attenuation rate	Learning rate
0	0.9	0.12
10	0.9	0.08
20	0.9	0.086
30	0.9	0.071
40	0.9	0.062
50	0.9	0.058

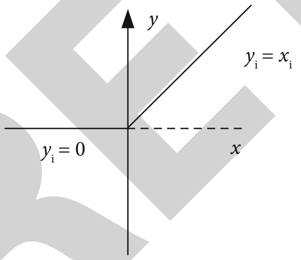


Figure 2: The ReLU activate function.

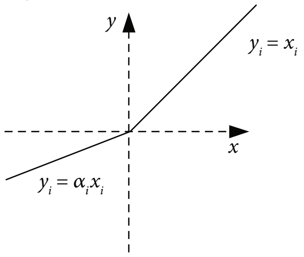


Figure 3: The LeakyReLU activation function.

through the index value. The training set, validation set, and test set are divided into 60%, 20%, and 20% in a chronological order. All models use Adam algorithm as parameter optimizer. The model parameters are determined using a grid search algorithm. The range of the learning rate is mainly [0.001, 0.0005, 0.0001, 0.00005, 0.00001], the range of the L2 penalty coefficient is mainly [0.0001, 0.00001, 0.000001, 0.0000001], the dimension of the embedding vector is set to 50 by default unless otherwise specified, and the batch size during training, for 2048, individual models may require finer-grained parameter tuning. The model adopts the early stopping method, and the model training will be terminated when the model has not significantly improved on the validation set after 3 iterations.

4. Analysis of Experimental Results

In order to authenticate the prediction ability and correctness of the HCN module, the HCN module is split out for experiments. The results are shown in Table 3. Among them, the DNN means that only the deep model is used, and Cross network means that only the cross-feature part in the DCN model is used. In order to make up for the lack of linear features, the Cross network and HCN models here also include linear LR modules. The value after the symbol "|" indicates the number of layers used. It can be seen that when the number of layers of HCN is 1, then the model structure is only one more dot product normalization operation than FvFM, but its effect is still a good improvement. Since, the effect of numerical magnitude recovery is not very large when the number of layers is low, this may be mainly due to the nonlinear factors brought by dot product normalization. When taking a better number of layers, compared with the Cross network in the DCN model, the HCN model has achieved greater advantages. This is evident from the

Table 2: Detailed statistics of each discrete dataset.

Data set	Number of samples	Number of numeric fields	Number of discrete domains	Number of features after processing
Criteo	44640677	14	28	121632
Conversion	16764869	12	9	33651
Avazu	41448978	0	24	112895

Table 3: High-order cross-model HCN experimental results AUC.

Model	Criteo	Conversion	Avazu
FvFM	0.7684|1	0.8418|1	0.7442|1
Cross network	0.7915|3	0.8415|3	0.7376|3
DNN	0.8124|3	0.8442|3	0.7468|3
HCN	0.8156|1	0.8436|1	0.7452|1
HCN	0.8094|3	0.8445|2	0.7486|3

accuracy of the HCN and other model as discussed later in this section.

The feature crossover mechanism of the HCN model still retains the domain relationship, and the crossover operation is more in line with the logical "and" thinking, unlike the Cross network model, it is only a matrix mapping of the overall splicing features, and only the highest-level feature information is output. Compared with the DNN model, the model effect of the HCN model is not too bad, and there are advantages and disadvantages in the data set. Because these two modules use different forms of high-level feature information, they have both certain accuracy and certain differences in the model effect. Therefore, when the two modules are jointly trained, the feature information of different perspectives is complementary, and the model is not affected. The improvement of decision-making is very obvious, and the mechanism of action is similar to ensemble learning in machine learning.

In order to examine and evaluate the influence of the numeral quantity of layers used by the HCN on the model effect, experiments are carried out on different layers, and the consequences are displayed in Figure 4. Through investigating these values, this can be found that when the numeral amount of layers is from 1 to 2, then the AUC of the model is improved significantly. At this time, the crossover order equivalent to the feature has been raised from order 2 to order 3-4. However, when the number of layers exceeds 3, then the effect of the model does not decrease, but there is basically no upward trend. This also means that a too high crossover order will not produce large fluctuations in the model performance. Similarly, when the number of domains is quite and potentially large, then the number of parameters reduced by using a smaller number of layers is still considerable.

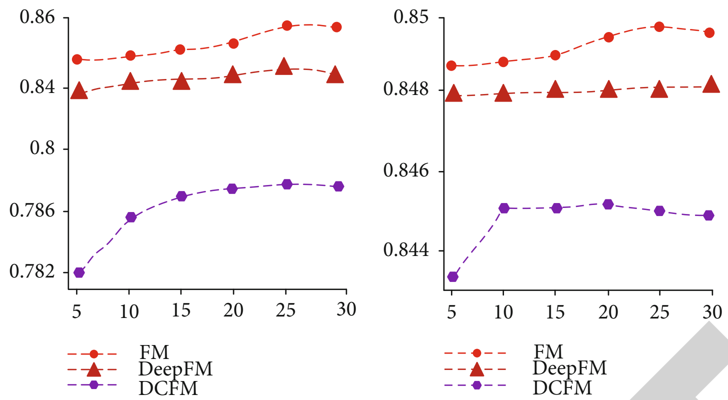
Figure 4 compares the AUC variation curves of the models under different embedding vector dimensions. Observing the trend of the curve, the factorization model (FM) is more sensitive to the value of the embedded vector dimension, and the model effect increases with the increase of the embedded vector dimension, until the value of 25 or later, the improvement is no longer obvious. On the contrary, the DNN-based models DeepFM and DCFM do not have high requirements on the dimension of the embedding vector. It can be seen that the embedding vector of different sizes does not have a great impact on the model. This may be due to the fact that the factorization class model belongs to the shallow model and needs more parameters to represent the semantics of the sparse features, while the deep neural network has a deeper network structure and can still learn high-level from the embedding vector of smaller dimensions and semantic features.

For both datasets, Figures 5 and 6 illustrates the accuracy of the various approaches against the suggest HCN model. For both datasets, we observed that the HCN model is significantly more accurate than the other approaches. The RMSE and MAPE values also demonstrate the supremacy of the HCN model.

5. Conclusions and Future Work

Faced with the problem of insufficient research on constructing high-order cross eigenvalues, this paper solves this problem by retaining the underlying FM structure and recombining cross eigenvalues into new sparse eigenvectors. Then repeat the steps of cross sparse eigenvectors of eigenvalues to obtain cross eigenvalue information of different orders. In order to avoid the problem that the magnitude of the output value is too small caused by multiple vector dot products, a dot product normalization operation is designed to effectively carry out gradient backpropagation. And the linear module and the depth module are combined to form an end-to-end learning of different levels of feature information. The effects are fully evaluated and compared on three public datasets, and experiments show that the algorithm has certain advantages over other prediction models. The simulation outcomes indicate that the suggested HCN approach has noble adaptability and high correctness in forecasting the click-through rate of marketing advertisements. We observed that this improvement, in terms of predictions precisions and accuracies, can be as high as 17.52% higher than the deep neural network (DNN) method and 10.45% higher than the factorization network (FM) approach.

In the future, advanced deep learning techniques like deep neural networks (DNNs) must be taken into account to boost the exactness of the prediction conclusions. The time-consuming training procedure for learning algorithms has the potential to reduce the system's overall performance. We will, in the near future, contemplate separating the training and prediction parts across the edge-cloud structural design for the purpose that the training could take place at the distant cloud, which normally has a lot of computational assets and power. Contrary to this, the prediction fragment of the procedure would execute on the cutting edge that, subsequently, will significantly shorten the system's response and processing or execution times. Furthermore, more



(a) Criteo

(b) Conversion

Figure 4: The effect of different embedding vector dimensions on the model.

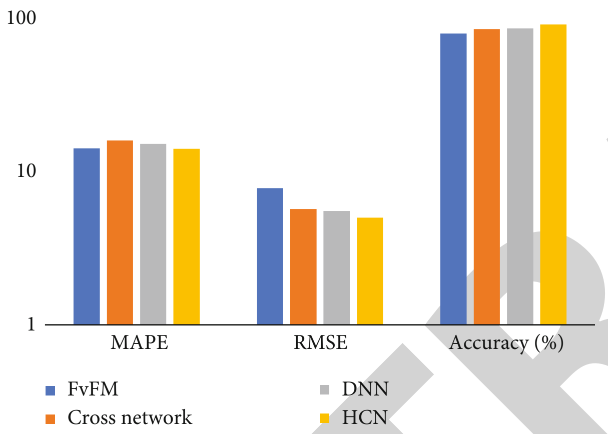


Figure 5: Comparison of various approaches in terms of accuracy (Criteo).

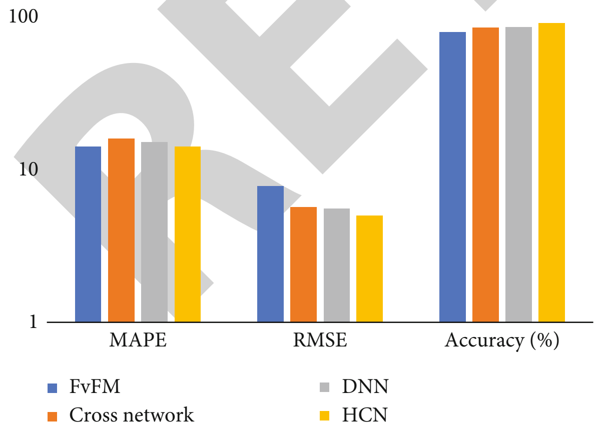


Figure 6: Comparison of various approaches in terms of accuracy (Conversion).

robust and faster deep learning approaches will be developed to improve the accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Authors' Contributions

Mo Li and WeiSheng Sun have contributed equally to this work and share first authorship.

References

[1] H. B. McMahan, G. Holt, D. Sculley et al., "Ad click prediction: a view from the trenches [C]," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, United States, 2013.

[2] Z. Aoying, Z. Minqi, and G. Xueqing, "Computational advertising: web comprehensive application with data as the core [J]," *Chinese Journal of Computers*, vol. 34, no. 10, pp. 1805–1819, 2011.

[3] Z. Zhiqing, Z. Yong, and X. Xiaoqin, "Research on advertising click-through rate prediction technology based on feature learning [J]," *Chinese Journal of Computer*, vol. 39, no. 4, pp. 780–794, 2016.

[4] A. Agarwal, O. Chapelle, M. Dudík, and J. Langford, "A reliable effective terascale linear learning system [J]," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1111–1133, 2014.

[5] K. Gai, X. Zhu, H. Li, K. Liu, and Z. Wang, "Learning piecewise linear models from large scale data for ad click prediction [J]," 2017, arXiv preprint arXiv:1704.05194.

[6] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*, pp. 995–1000, Sydney, NSW, Australia, 2010.