Wireless Communications and Mobile Computing

Artificial Intelligence-Powered Systems and Applications in Wireless Networks

Lead Guest Editor: Wenzhong Li
Guest Editors: Xiao Zhang, Lin Wang, and Yanjie Fu

Artificial Intelligence-Powered Systems and Applications in Wireless Networks

Wireless Communications and Mobile Computing

Artificial Intelligence-Powered Systems and Applications in Wireless Networks

Lead Guest Editor: Wenzhong Li
Guest Editors: Xiao Zhang, Lin Wang, and Yanjie Fu

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5871114, 11 pages
https://doi.org/10.1155/2021/5871114

Research Article

LB-DDQN for Handover Decision in Satellite-Terrestrial Integrated Networks

Dong-Fang Wu,[1,2] **Chuanhe Huang**,[1,2] **Yabo Yin**,[1,2] **Shidong Huang**,[1,2] **M. Wasim Abbas Ashraf**,[1,2] **Qianqian Guo**,[3] **and Lin Zhang**[4]

[1]*School of Computer Science, Wuhan University, Wuhan 430072, China*
[2]*Hubei Luojia Laboratory, Wuhan 430072, China*
[3]*School of Information Engineering, Zhengzhou Institute of Finance and Economics, Zhengzhou 450053, China*
[4]*Wuhan Maritime Communication Research Institute, Wuhan 430072, China*

Correspondence should be addressed to Chuanhe Huang; huangch@whu.edu.cn

Received 3 June 2021; Revised 22 October 2021; Accepted 15 November 2021; Published 8 December 2021

Academic Editor: Yanjie Fu

The frequent handover and handover failure problems obviously degrade the QoS of mobile users in the terrestrial segment (e.g., cellular networks) of satellite-terrestrial integrated networks (STINs). And the traditional handover decision methods rely on the historical data and produce the training cost. To solve these problems, the deep reinforcement learning- (DRL-) based handover decision methods are used in the handover management. In the existing DQN-based handover decision method, the overestimates of DQN method continue. Moreover, the current handover decision methods adopt the greedy strategy which lead to the load imbalance problem in base stations. Considering the handover decision and load imbalance problems, we proposed a load balancing-based double deep Q-network (LB-DDQN) method for handover decision. In the proposed load balancing strategy, we define a load coefficient to express the conditions of loading in each base station. The supplementary load balancing evaluation function evaluates the performance of this load balancing strategy. As the selected basic method, the DDQN method adopts the target Q-network and main Q-network to deal with the overestimate problem of the DQN method. Different from joint optimization, we input the load reward into the designed reward function. And the load coefficient becomes one handover decision factor. In our research, the handover decision and load imbalance problems are solved effectively and jointly. The experimental results show that the proposed LB-DDQN handover decision method obtains good performance in the handover decision. Moreover, the access of mobile users becomes more balancing and the throughput of network is also increased.

1. Introduction

As one important network of the future wireless networks, the STIN [1] offers mobile users communication services with wide coverage, high reliability, and low delay. This new integrated networks consist of terrestrial segment network and satellite segment network, which provides the future smart city with a ubiquitous network. Because of the increase of user's equipment and the mobility of users, the mobile users need to connect the optimal candidate cell to continue the communication. But the overlapped area, the access limitation of base station, and the random mobility of users make the handover more complex. How can mobile users decide the optimal candidate cell? Which decision factor is the key factor? Meanwhile, mobile users also need to deal with the handover decision problem in data transmission services. This research problem can directly influence the handover performance and Quality of Service (QoS) of mobile users in wireless networks. The handover management technology includes three steps: (1) information collection, (2) handover decision, and (3) handover execution [2]. Aiming at the frequent handover, handover failure, and load imbalance in the terrestrial segment of STIN, combining with the load balancing strategy, a load

balancing-based double deep Q-network (LB-DDQN) handover decision method is proposed.

The developments of handover decision algorithm attract more attention from academia and industry. The traditional handover decision methods include the simple additive weighting method, TOPSIS, decision function method, and Q-learning. These traditional methods focus on the measurement report, handover threshold and decision function, which relay on the priori knowledge and produce training cost. And the latest method is deep reinforcement learning (DRL) which combines the feature analysis ability of deep learning and decision-making ability of reinforcement learning. The DRL method effectively resolves the mobility of users and dynamics of networks. In the existing DRL methods, the value function-based DRL and the policy gradient-based DRL are the core of the fundamental approach and hot spots. The policy gradient-based DRL method is widely used in the Markov decision process with continuous action space. The DQN-based method is used in action space with discrete low dimension. And the researched handover decision problem in STIN is one typical deterministic discrete decision problem. And the DDQN method adopts the target Q-network and main Q-network to deal with the overestimate problem of DQN method. Therefore, we select the improved DDQN method to train the handover decision process.

Figure 1 shows that users' mobility patterns and distribution differences lead to the load imbalance in the overlapped area of MBS1 and MBS2. And the current handover decision methods adopt the greedy strategy which also lead to the load imbalance problem in base stations. In Figure 1, this arrow expresses the moving direction of LEO satellites along the periodic trajectory in the orbit. These vital decision factors affect the handover decision and lead to frequent handover and decreased network throughput. Recently, few studies pay attention to both handover decision and load balancing problems. Moreover, the unevenly distributed mobile users are forced to switch to the base stations with low load in the handover-based load balancing method. Unlike from this method, the SINR, delay, and load coefficient were selected as the handover decision factors in the LB-DDQN handover decision method. Combining with the load balancing strategy, the improved DDQN method constructed the Markov decision model to realize the handover decision. In our research work, the handover decision and load imbalance problems are solved effectively and jointly. The proposed method has good performance of handover and meets the demands of load balancing. In this research, our contributions are summarized as follows:

(1) We proposed the LB-DDQN handover decision method in STIN to deal with the frequent handover and load imbalance. The handover decision problem in the terrestrial segment of the STIN was resolved. Furthermore, the feasibility of the LB-DDQN handover method was proved by the experiments and analysis

(2) We constructed the load balancing strategy, including load coefficient, the reward of load, and load balancing evaluation function. The reward of load analyzed the load's influence on the handover in the LB-DDQN method. With the help of the load balancing evaluation function, the load condition of the whole network was intuitively evaluated

(3) We selected the load coefficient as the handover decision factor. In the training process of handover decision, the load balancing was also considered. The handover decision and load balancing were both realized. In our research work, the handover decision and load imbalance problems are solved effectively and jointly

The rest of this paper is organized as follows. The related works of handover decision are surveyed in Section 2. The system model is described in Section 3. The LB-DDQN handover decision method is proposed in Section 4. Simulation setups and results of experiments are provided in Section 5. Finally, Section 6 concludes this paper.

2. Related Work

The existing handover decision methods in the network can be divided into four categories [3]: the decision function method, the multiattribute decision method, the context-aware decision method, and the artificial intelligence decision method. In [4], a multiattribute decision method computed the weights of decision factors. The simple additive weighting method (SAW), the technique for order of preference by similarity to ideal solution (TOPSIS), and the grey relational analysis method (GRA) were adopted to select the optimal candidate cell base station. In [5], the analytic hierarchy process method (AHP) was used to obtain the weights of decision factors, and the orders of candidate base station were computed by the SAW method. In [6], the AHP method obtained the weights of decision factors, and the GRA method ranked the candidate cell base stations. In [7], combining with the property normalization and weight calculation, the improved multiattribute decision method ranked the candidate cell base stations. In [8], the designed decision maker performed the network selection and handover decision. In [9], a received signal strength indicator- (RSSI-) based fuzzy logic method executed fast handover and seamless handover. Considering the frequent handover and ping-pong effect, a predicted signal to interference plus noise ratio- (SINR-) based handover decision method was proposed [10]. In [11], a speed-aware-based handover decision method was proposed to deal with the influence of the handover process on the network throughput in the two layers' cell network. This handover decision method failed to choose the best candidate base station but the base station which kept the maximum service cycle. This selection makes sure of the cooperation between base stations and the elimination of interference. In [12], the improved fuzzy TOPSIS method reduced the scope of the candidate base stations, which increased the throughput of the network. In [13], a handover decision method using fuzzy logic and combinatorial fusion was proposed. According to the RSSI, data rate, and network delay, the handover was predicted. Combining with the TOPSIS, the proposed handover decision method

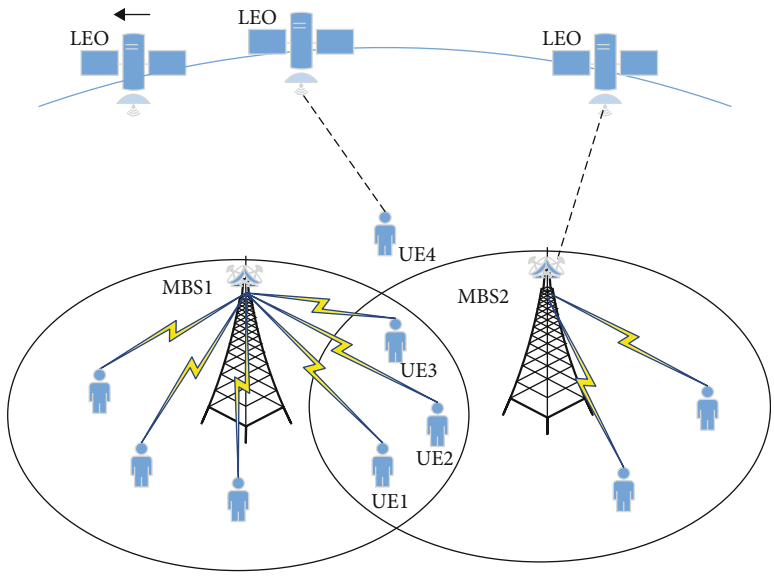


Figure 1: The scenario of handover decision and load imbalance in STIN.

assisted the mobile users to selecting the proper candidate base station [14]. In [15], combining with the improved competitive auction technique, the quality of uplink and downlink, and load factor, and the fast game-based handover decision method was proposed. The stochastic geometry analysis method was proposed [16]. By this method, the influence of different network topologies on handover decisions was analyzed. Moreover, the approximated handover number was estimated. In [17], an analysis framework of handover based on stochastic geometry was proposed to analyze the number of base stations, triggering time, and mobility patterns of users. The authors proposed a handover decision method based on fuzzy logic for saving the energy of mobile devices in an integrated LTE and Wi-Fi network [18]. And the traditional handover decision methods rely on the historical data and produce the training cost. The decision function method, the multiattribute decision method, and the context-aware decision method depend on the collected information about networks and users. This information plays an essential role in the handover decision. However, the collection of these vital information takes too much time. The delayed information and the dependence of prior knowledge lead to the frequent handover and load imbalance in the STIN.

The test of the reinforcement learning method on satellites designed by NASA proved that the artificial intelligence decision method had a good performance, and the deployment of this method was feasible [19]. In [20], the Markov handover decision model was constructed, and the hybrid vertical handover decision method was proposed. In [21], a Markov decision process- (MDP-) based handover decision method was proposed to optimize the QoS of network communication. In [22], considering the channel quality and QoS of communication, a reinforcement learning-based handover decision method was proposed from the point of user number. Google DeepMind team proposed the deep reinforcement learning method (DRL) which was evaluated in the Atari 2600, and this method achieved the excellent performance [23]. This new artificial intelligence method was used in communications and networking to deal with dynamic network access, data rate control, wireless caching, data offloading, and resource management [24]. In [25], a multiagent DRL method was proposed to resolve the distributed handover management problem. In this method, considering the cost factor, the user was modelled as an agent and the handover decision was optimized. In [26], the mobility patterns of users were classified, and the asynchronous multiagent DRL method was used in the handover decision process. In [27], the convergence speed and accuracy of the Q-network were optimized by the evolution strategy. The reinforcement learning-based handover decision method had good decision-making ability and handover performance. However, the state space, action space, and reward function of the different network scenarios need to be adjusted, which leads to the performance fluctuation. Moreover, the reinforcement learning-based handover decision method need to search the Q-table efficiently. This kind of method is suitable for the discrete state space problem. Replacing the Q-table by the neural network, the DRL-based handover decision method is good at dealing with the continuous state space problem. Therefore, our research adopts the improved DRL method to train the handover decision process.

3. System Model

3.1. Network Model and Problem Formulation. The terrestrial segment of the STIN makes up of M macro cells and N mobile users. In the satellite segment, there is always one LEO satellite that provides the satellite communication service. The mobile users select base station or LEO satellite to transmit data. Our research focused on the handover decision problem in the terrestrial segment of STIN. The network time T is divided into many time slots. In each time

slot t, every mobile user selects the optimal target base station from the M candidate base stations. When the mobile users move out of the range of network or the network time T is end, the state of the user update to tend, $T = \{t_0, t_1, \cdots, t_{\text{end}}\}$ includes many discrete handover decision scenarios. After the construction of the Markov handover decision model, the handover decision process is optimized by the neural network.

The handover decision process of mobile users is modelled as the discrete Markov decision process, expressed by $<S, A, \pi, r, \gamma>$, where S is the discrete state space of the network, which is composed of network parameters. The parameter A is the action space that is composed of the candidate base stations set. The parameter $\pi : S->A$ shows that the action A can be determined by the state S. The function of reward $r : S \times A->R$ computes the positive or negative rewards from the network environment. The discount coefficient γ describes the value of future reward. Figure 2 shows that the agent obtains the recent state s_t in each time slot t. The action a_t is determined by the strategy π. After receiving the parameter of action a_t, the network environment returns a reward r_t which shows whether the action is proper. Then, the state of network environment is update to s_{t+1}.

The proposed LB-DDQN handover decision method is aimed at maximizing the total reward of the handover decision. In the interaction of agent and environment, the discounted total reward G_t is defined as

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma \bullet R_{t+2} + \gamma^2 \bullet R_{t+3} + \bullet\bullet\bullet, \quad (1)$$

where R is the immediate reward of handover. The discount coefficient of future reward is named γ. According to Equation (1), the Bellman operator updates the action-value function $Q(s_t, a_t)$.

$$Q(s_t, a_t) = E[R_{t+1} + \gamma \bullet Q(s_{t+1}, a_{t+1})], \quad (2)$$

where s_t and a_t are state and action, respectively, in time slot t. According to Equation (2), the optimal Bellman operator is defined as

$$Q^*(s_t, a_t) = E\left[R_{t+1} + \gamma \bullet \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})\right]. \quad (3)$$

Equation (3) describes that the maximum value of the action-value function $Q(s_{t+1}, a_{t+1})$ is computed. In the LB-DDQN handover decision method, the neural network is used to estimate the action-value function.

3.2. State Space and Action Space. In each time slot t, the size of candidate base station set for mobile users is M. We select the SINR, delay, and load coefficient as the decision factors. As for the candidate base station i, the state space is expressed as $c_i(t) = \{\text{SINR}_i(t), D_i(t), L_i(t)\}$. The total network state space is defined as $c(t) = (c_0(t), c_1(t), c_2(t), \cdots, c_{M-1}(t))$. The public interface X2 shares the load information of each base station. Moreover, the SINR and the delay information are obtained by the regular measurement reports. These selected decision factors assist in the handover decision.

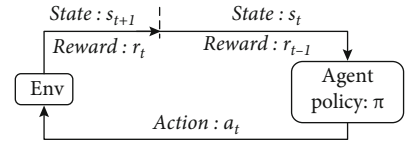


Figure 2: The framework of the reinforcement learning.

The action of handover for the mobile user is expressed by parameter a. And the action space is made up of all the indexes of the candidates base station, expressed as $A = \{0, 1, 2, \cdots, M-1\}$. According to the ε-greedy strategy, in each time slot t, the mobile user selects one proper base station to connect. When the action parameter $a = 0$, the base station whose index is 0 is selected.

3.3. Reward Function. Considering the selected network parameters in state space, the reward function is defined as

$$R_t = \sum_x w_x r_{x_i(t)}, \quad (4)$$

where the normalized reward of the parameter x for the candidate base station i in time slot t was expressed by $r_{xi(t)}$. The variable w_x is the weight coefficient of the parameter x ($0 \le w_x \le 1$). These weights are computed by the AHP method [6]. The load reward is computed from the load coefficient. The SINR can be obtained from the measurement report of base station. As the positive parameters such as SINR and load, the reward function is defined as

$$r_{x_i(t)} = \begin{cases} 1, & x_i(t) \ge X_{\max}, \\ \dfrac{x_i(t) - X_{\min}}{X_{\max} - X_{\min}}, & X_{\min} < x_i(t) < X_{\max}, \\ 0, & x_i(t) \le X_{\min}, \end{cases} \quad (5)$$

where the variables $X_{\max}$ and $X_{\min}$ are the maximum value and minimum value of network parameter x. The network parameter for candidate base station i in time slot t is expressed as $x_i(t)$ which is evaluated between [0,1]. As for the negative parameter, such as delay, the reward function is defined as

$$r_{x_i(t)} = \begin{cases} 1, & x_i(t) \le X_{\min}, \\ \dfrac{X_{\max} - x_i(t)}{X_{\max} - X_{\min}}, & X_{\min} < x_i(t) < X_{\max}, \\ 0, & x_i(t) \ge X_{\max}. \end{cases} \quad (6)$$

4. LB-DDQN-Based Handover Decision Method

4.1. Traditional Handover Decision Methods. The traditional handover decision methods include SAW [4], TOPSIS [17], and Q-learning [28]. Compared with the DRL handover decision method, these traditional methods depend on the

measurement report and prior knowledge. The traditional handover decision methods are unsuitable for the dynamically changing network environment. The load balancing problem is also not fully considered. The SAW method computes the order of the candidate base station. The sum of normalized parameters is defined as

$$\mathrm{Add}_i = \sum_{j=1}^{n} V_{ij} * \beta_j, \tag{7}$$

where the variable Add_i express the sum of normalized network parameter in the candidate base station i. The variable β_j is the weight of the parameter j. The variable V_{ij} is the normalized value of parameter j in the candidate base station i, where $i = 0, 1, 2, \cdots, M-1$ and $j = 1, 2, 3$. The TOPSIS [17] selects the optimal candidate base station. The Euclidean distance is defined as

$$\begin{aligned} U_i^* &= \sqrt{\sum_{j=1}^{n} \left(V_j^* - V_{ij}\right)^2}, \\ U_i' &= \sqrt{\sum_{j=1}^{n} \left(V_j' - V_{ij}\right)^2}, \end{aligned} \tag{8}$$

where U_i^* and U_i' are the Euclidean distances between V_{ij} and the optimal solution set $\{V_1^*, V_2^*, \cdots, V_n^*\}$ and the worst solution set $\{V_1', V_2', \cdots, V_n'\}$, respectively. We use the variable V_j^* and V_j' to express the maximum value and the minimum value of parameter j, respectively. And the closeness efficient C_i^* is defined as

$$C_i^* = \frac{U_i'}{U_i' + U_i^*}, \tag{9}$$

when the variable C_i^* close to 1 and the U_i^* is small which means the Euclidean distance between the candidate solution and optimal solution is small. The update of the action-value function in the Q-learning method is defined as

$$Q(s_t, a_t) = Q(s_t, a_t) + \eta \bullet \left(R_{t+1} + \gamma \bullet \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right). \tag{10}$$

4.2. Load Balancing Strategy. One base station services the fixed number of mobile users at the same time. When the number of connected users exceeds the upper limit of load or the mobile user cannot connect to the target base station, the handover request is failed. The proposed LB-DDQN handover decision method realizes the handover decision and load balancing simultaneously by the load balancing strategy. The limited resource block and unevenly distributed users fail the handover request. By the LB-DDQN method, the number of handover failure is effectively decreased. The load coefficient, load reward, and load balancing evaluation function are presented in the designed load balancing strategy. In time slot t, the load coefficient of base station i expressed by $L_{i,t}$ $(0 \le L_{i,t} \le 1)$ which is defined as

$$L_{i,t} = \frac{\mathrm{UEnum}_{i,t}}{\mathrm{Tnum}_i}. \tag{11}$$

The variable $L_{i,t}$ expresses the number of serving users. Assume that one mobile user connects up to one base station and occupies one resource block. In time slot t, the variable $\mathrm{UEnum}_{i,t}$ is the number of serving users. The variable $\mathrm{Tnu\,m}_i$ expresses the total number of resource blocks. When the load coefficient of the base station increases, the number of serving user increases. At this time, the load reward of the base station decreases, and the probability of handover selection decreases. On the contrary, when the load coefficient of the base station becomes small, the available source blocks and load reward become large. This base station is more possibly selected as the optimal candidate base station. In time slot t, the load reward of the base station i is defined as

$$\mathrm{HO\_reward}_{i,t} = \frac{1}{L_{i,t} + 1}, \tag{12}$$

where the load reward of the base station i in time slot t is expressed by $\mathrm{HO\_reward}_{i,t}$. Its value range is [0.5,1]. In time slot t, the load balancing evaluation function is defined as

$$\mathrm{LB}_t = \sqrt{\sum_{i \in M} \left(L_{i,t} - \frac{\sum_{i \in M} L_{i,t}}{M}\right)^2}, \tag{13}$$

where LB_t is the value of load balancing function in time slot t. When the value of this variable is bigger, the distribution of mobile users is more imbalanced. When the variable LB_t close to 0, the load of the base stations is balanced. The operation $\sum_{i \in M} L_{i,t}/M$ obtains the average load coefficient of all base stations.

4.3. Implementation of LB-DDQN. The DRL handover decision method adopts the neural network to estimate the optimal value of the action-value function. In the training process of handover decision, the normalized parameters of state space are regarded as the input of the neural network, and the optimal value of the action-value function is output.

$$Q(s_t, a_t; \theta) \approx Q^*(s_t, a_t). \tag{14}$$

In [23], the update of action-value function in the DQN method is defined as

$$Q_m(s_t, a_t) = Q_m(s_t, a_t) + \eta * \left(R_{t+1} + \gamma \bullet \max_a Q_t(s_{t+1}, \mathrm{a}) - Q_m(s_t, a_t)\right), \tag{15}$$

where Q_m is the action-value function of main Q-network and Q_t is the action-value function of target Q-network. We proposed the improved LB-DDQN handover decision method. When the maximum value of Q_m is obtained,

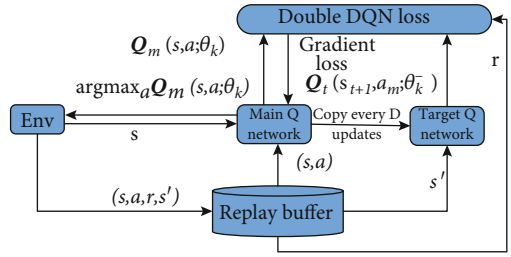


Figure 3: The framework of DDQN method.

the handover action a_m corresponding to the optimal Q_m is determined. The update of action-value function is defined as

$$a_m = \arg \max_a Q_m(s_{t+1}, a),$$
$$Q_m(s_t, a_t) = Q_m(s_t, a_t) + \eta * (R_{t+1} + \gamma \bullet Q_t(s_{t+1}, a_m) - Q_m(s_t, a_t)). \tag{16}$$

The loss function of DDQN method is the difference value between the target value y and the estimated action-value function $Q_m(s_t, a_t, \theta_k)$. The loss function is defined as

$$y = \begin{cases} R_{t+1}, & \text{if } s_{t+1} \text{ is end,} \\ R_{t+1} + \gamma \bullet Q_t(s_{t+1}, a_m; \theta_k^-), & \text{others,} \end{cases} \tag{17}$$
$$L_k(\theta_k) = E\left[(y - Q_m(s_t, a_t; \theta_k))^2\right].$$

In the training process of handover decision, the loss function returns the gradient loss to update the parameters of main Q-network at each iteration. With the updates of the parameters, the loss value of loss function decreases. And the performance of the handover becomes better. The loss function of the DDQN method is optimized by the stochastic gradient descent method. The gradient of loss function is defined as

$$\nabla_{\theta_k} L(\theta_k) = E\left[(y - Q_m(s_t, a_t; \theta_k)) \nabla_{\theta_k} \bullet Q_m(s_t, a_t; \theta_k)\right]. \tag{18}$$

As Figure 3 shows, the DDQN handover decision method adopts the main Q-network Q_m and the target Q-network Q_t. These two neural networks are initialized with the same network parameters. In the main Q-network Q_m, the network parameters are updated at each iteration. The main Q-network Q_m is used to estimate the value of action-value function. After every D steps, the network parameters of target Q-network are updated by the parameters of main Q-network. $\theta_k^- = \theta_k$. The target Q-network Q_t is used to compute the expected value of action-value function.

In the training process of handover decision, the experimental data is saved in the replay buffer. By the experience replay method and the small batch sampling method, the randomly sampled data is used as input data to train the parameters of Q-network. By the ε-greedy strategy, the exploration and the exploitation operations of the optimal handover action are realized. The detailed steps of the LB-DDQN handover decision method are described as Algorithm 1.

5. Simulation Results and Discussions

5.1. Simulation Environment. This research makes sure of the handover performance and load balancing requests simultaneously. A PC carries out the experiments with 3.2 GHz quad-core i5-1570 and 16GB of RAM. The OS is win 10, 64 bits, and the simulation platform is python3. Figure 4 shows that it is 2290 meters long and 1800 meters wide in the virtual town scenario. In this network area, there is one LEO satellite that provides 24-hour communication services. It includes 31 base stations whose communication range is 500 meters. Assume that the base station bandwidth is 10 MHz and the upper limits of connected users for the base station are 50. One user only occupies up to one resource block, and the bandwidth of each subchannel is 180 kHz. The starting point of the mobile user is randomly selected from 11 crossings. The speed of the mobile user is randomly selected from 5 km/h, 25 km/h, 50 km/h, 70 km/h, and 120 km/h. The mobile user is moving at a constant speed in straight lines. The number of mobile users is 50, 100, 200, and 200, respectively.

5.2. Simulation Parameters. The handover rate, handover failure rate, and throughput of the network are used to evaluate the handover performance. The simulation parameters are illustrated in Table 1.

The handover rate and handover failure rate are defined as

$$\text{HOR} = \frac{N_{\text{HO}}}{N_{\text{total}}},$$
$$\text{HOF} = \frac{N_{\text{re}} - N_{\text{HO}}}{N_{\text{total}}}, \tag{19}$$

where the variable N_{HO} is the number of successful handover and the variable N_{total} is the total number of handover decision. The N_{re} expresses the number of handover requests. The range of HOR and HOF is [0, 1]. The network parameter SINR is defined as

$$\text{SINR} = 10 \bullet \log\left(\frac{P_S}{P_I + P_N}\right), \tag{20}$$

where the variable P_S is the effective power and the variable P_I is the interference power. And the variable P_N is the noise power. The throughput of network Th is defined as

$$\text{Th} = W * \log_2\left(1 + \frac{P_S}{P_I + P_N}\right) \tag{21}$$

where the variable W is the bandwidth of the subchannel.

5.3. Simulation Results

5.3.1. Average Handover Number of User. As Figure 5 shows, the handover numbers of the LB-DDQN method are

Input: Iteration number NUM_EPISODES, step number MAX_STEPS, node number node_num, measurement information SINR and delay, length of update step D.
Output: Handover decision matrix $\boldsymbol{A}$.
1: Initialize action-value function Q, replay buffer B and handover decision matrix $\boldsymbol{A}$. The initialized parameters of the main Q-network and target Q-network are consistent. $\theta_k^- = \theta_k$.
2: **for** i=1, NUM_EPISODES **do**
3: **for** j=1, MAX_STEPS **do**
4: **for** k=1, node_num **do**
5: According to Eq. (4, 5, 6), the immediate reward $r_{j,k}$ is computed.
6: According to Eq. (11, 12, 13), the load coefficient $L_{i,t}$ and load reward HO_reward are obtained. Construct the sequences of state $s_{j,k}$ include: SINR, delay and HO_reward.
7: By the ε-greedy strategy, the handover action $a_{j,k}$ corresponding to the state $s_{j,k}$ is determined. And the handover decision matrix $\boldsymbol{A}$ is updated.
8: Construct the next state $s'_{j,k}$, the experience data $(s_{j,k}, a_{j,k}, r_{j,k}, s'_{j,k})$ is saved in the replay buffer B.
9: Using experience replay and small batch sampling method, the randomly sampled data from the replay buffer B is produced and input the main Q-network Q_m. Then the action-value function $Q_m(s,a)$ is obtained.
10: According to the Eq. (16, 17), the action a_m corresponding to the maximum value of Q_m is obtained and input the target Q-network Q_t. And the action-value $Q_t(s'_{j,k}, a_m)$ is computed.
11: Adopt the stochastic gradient descent method, according to Eq. (18), the parameters θ_k of main Q-network are updated.
12: **end for**
13: Every D steps, the parameters of target Q-network are updated by the parameters of main Q-network. $\theta_k^- = \theta_k$.
14: **end for**
15: **end for**
16: Return the handover decision matrix $\boldsymbol{A}$.

Algorithm 1: LB-DDQN handover decision algorithm.

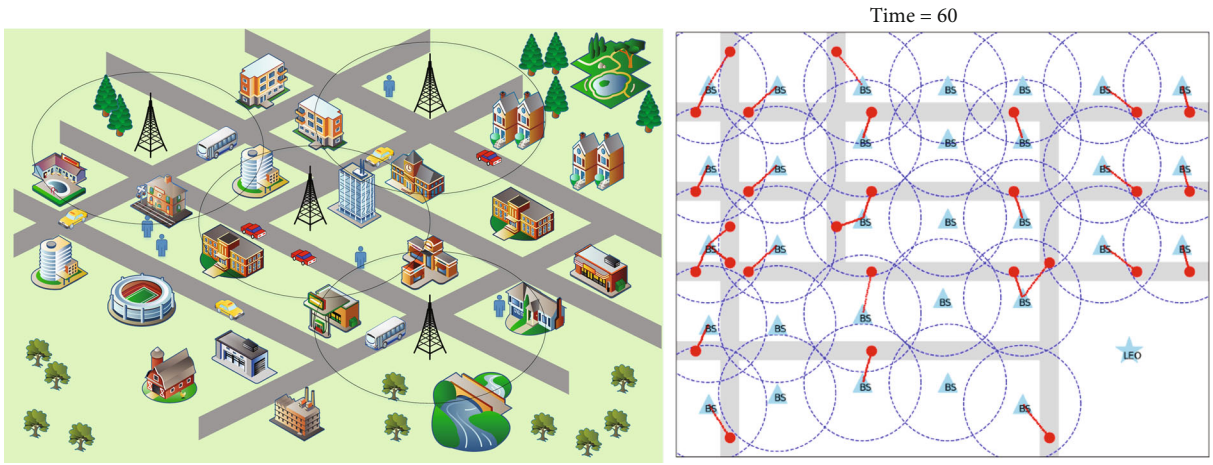


Figure 4: The scenario of the virtual town and the connection of mobile users at 60 second.

Table 1: Simulation parameters of the network.

Parameters	Values
Total number of BS	31
Bandwidth	10 MHz
Thermal noise	-174 dBm/Hz
Shadowing	8 dB
Cell transmit power	46 dBm
Path loss model	128.1 + 37.6 ∗ lg (d)
Number of UE	50, 100, 200, 300
UE speeds (km/h)	5, 25, 50, 70, 120
Duration of simulation	600 seconds

compared. The speeds of the user are set to 5 km/h, 25 km/h, 50 km/h, 70 km/h, and 120 km/h, respectively. The amounts of mobile users are set to 50, 100, 200, and 300, respectively. This figure assists in analyzing the influence of user speed and amount of users on handover performance. As we can see, when the speed of mobile user increases, the handover number of users gradually decreases. This is because in the network time T, when the user speed increases, the user is earlier entering the final state. And the decrease of the effective sampling points in the simulation leads to the decreased of handover decision number and handover times. In the virtual town scenario of the network, when the user speed is 120 km/h, and the length of the road is 1.8 km, the number of effective sampling points is 545. When the user speed

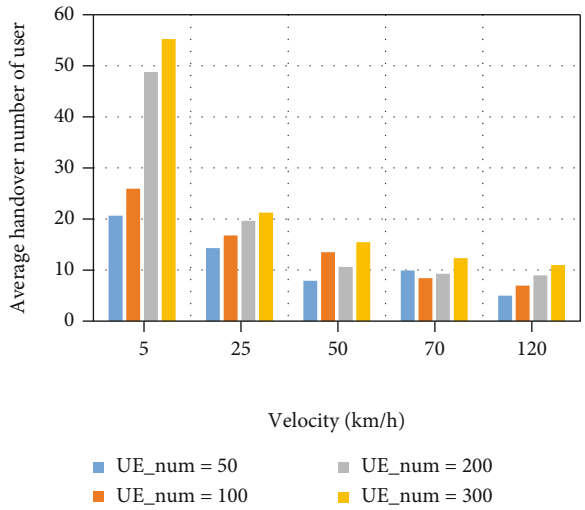


Figure 5: The handover number of LB-DDQN method with different user speeds.

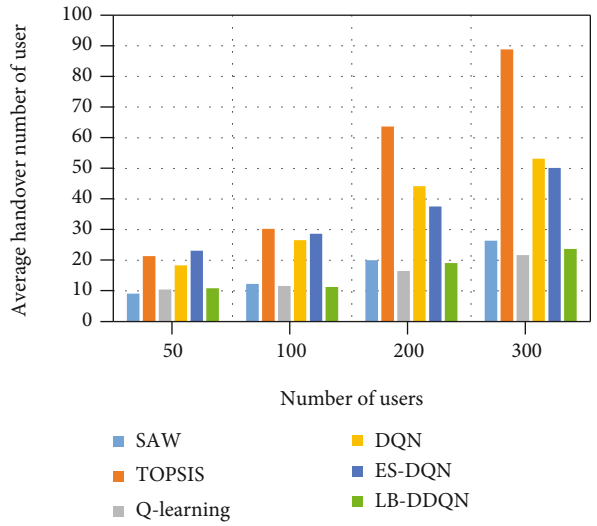


Figure 6: The handover performance of different handover decision methods.

changes to 5 km/h, the number of effective sampling points is 6000. Moreover, when the user speed is fixed, the increase of user number results in the increased of handover number. It is observed that the increase of mobile users leads to the increase of distribution difference, handover number, and interference signal. Meanwhile, handover management becomes more complex, because the SINR factor is also one of the handover decision factors, which leads to the increase of handover number.

As Figure 6 shows, the SAW [4], TOPSIS [17], Q-learning [28], DQN [27], and ES-DQN [27] handover decision methods are compared with the proposed LB-DDQN handover decision method in a different number of users. This figure shows the performance difference of these handover decision methods. As the increase of the number of users, the average handover number of user also increases. As for the traditional handover methods, the Q-learning-based handover method has the optimal handover performance. As for the artificial intelligence-based handover decision method, considering the load balancing factor as the decision factor, the proposed LB-DDQN method optimizes the handover decision process, and the average handover number of user decreases.

As we can see from Table 2, when the number of user is 50, the optimal handover decision method is the SAW method whose average handover number of a user is 9.1. The average handover number of the LB-DDQN method is 10.9. When the number of users is 100, the optimal handover decision method is the LB-DDQN method and the corresponding average handover number is 11.32. When the number of users is 200, the optimal handover decision method is Q-learning, and the average handover number is 16.53. And the average handover number of the LB-DDQN method is 19.07. When the number of user is 300, the optimal handover decision method is Q-learning, and the average handover decision method is 21.67. The average handover number of the LB-DDQN method is 23.74. The proposed LB-DDQN method has good handover, which is the same as the Q-learning. Moreover, the LB-DDQN method makes sure of the performance of handover and the continuity of data services.

Table 2: Average handover number of users for different handover decision methods.

UE_ Num	SAW	TOPSIS	Q-learning	DQN	ES-DQN	LB-DDQN
50	9.1	21.34	10.44	18.32	23.12	10.9
100	12.3	30.26	11.62	26.57	28.68	11.32
200	20.05	63.64	16.53	44.22	37.55	19.07
300	26.4	88.82	21.67	53.21	50.12	23.74

5.3.2. Handover Rate and Failure Rate. As Figure 7 shows, the handover rate and failure rate of different handover decision methods are compared. The number of user is 100. According to Equation (19), the handover rate and failure rate are computed. As for the handover rate, the optimal handover decision method is Q-learning. The handover rate of Q-learning is 0.0019. And the handover rate of LB-DDQN is 0.0021. As for the failure rate, the optimal method is the ES-DDQN method. Its failure rate is 0.0067. And the LB-DDQN is 0.007 which is better than Q-learning method. We find that the proposed LB-DDQN method has the good performance of handover rate and failure rate. Considering the load factor, the access of mobile users is more balanced, and the continuity of data services is enhanced. Compared with the DQN and ES-DQN methods, the LB-DDQN method decreases the handover rate, which makes sure of the request of handover. At the same time, by the load balancing strategy, the failure rate also decreases.

5.3.3. Throughput of Network. As Figure 8 shows, the throughput of the network for different handover decision methods are described. The number of users is 100. The optimal method is the proposed LB-DDQN handover

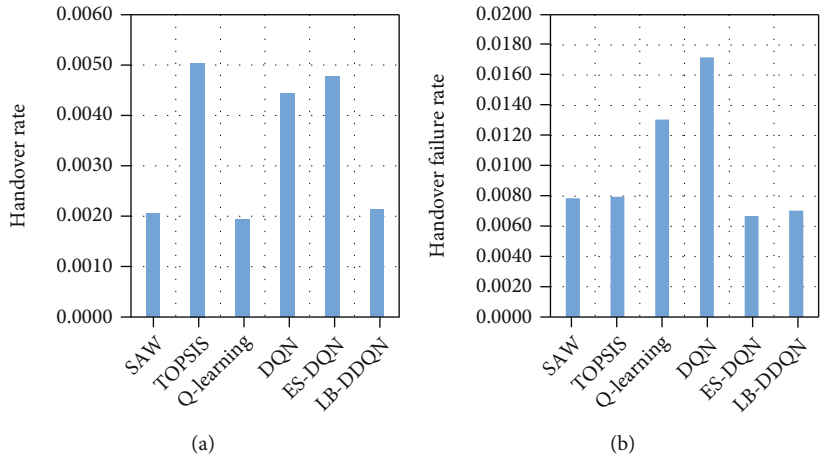


Figure 7: The handover rate (left) and failure rate (right) of different handover decision methods.

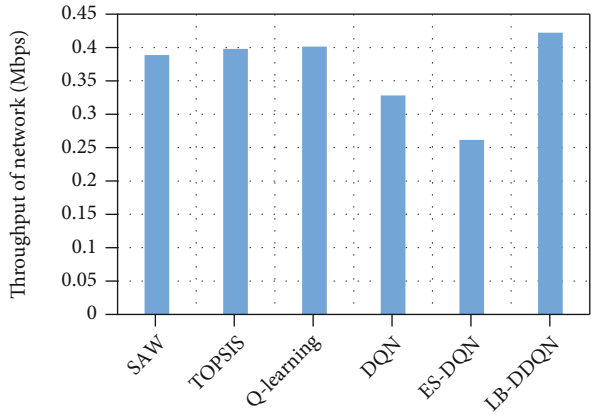


Figure 8: The throughput of network for mobile users.

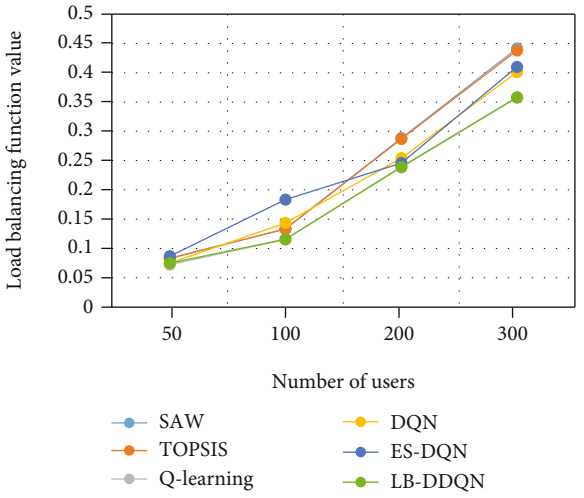


Figure 9: The values of load balancing function for different handover decision methods.

decision method whose throughput of the network is 0.4221 Mbps. The network throughput of the -learning method is 0.4012 Mbps. We find that by combining the load factor, the network throughput of the LB-DDQN method is higher than those of the others. Our load balancing strategy eliminates the effects of frequent handover, handover failure, and load imbalance.

5.3.4. Load Balancing Function Value. As Figure 9 shows, the evaluation of load balancing for these methods is described. The number of mobile users is 50, 100, 200, and 300, respectively. The load is smaller, and the distribution of mobile user is more balanced. As the number of users increases, the value of load balancing function increases. Moreover, the optimal method is the LB-DDQN handover decision method. This is because our method combines the load balancing strategy, and the load coefficient is also the decision factor.

As Table 3 shows, the detailed value of load balancing function is described. The number of users is 50, 100, 200, and 300, respectively. When the number of users is 50 and 100, respectively, the optimal method is the Q-learning method. The values of load balancing function are 0.0724 and 0.1154, respectively. When the number of users is 200 and 300, respectively, the optimal method is the LB-DDQN handover method. The values of load balancing function are 0.2388 and 0.3577, respectively. As the number of users increases, the difference in distribution of mobile users is more complex. The performance of the proposed load balancing strategy for the mobile user is good and proper.

Table 3: The values of load balancing function for different handover decision methods.

UE_ Num	SAW	TOPSIS	Q-learning	DQN	ES-DQN	LB-DDQN
50	0.0834	0.0832	0.0724	0.075	0.0868	0.075
100	0.1336	0.1333	0.1154	0.1436	0.1836	0.1158
200	0.2882	0.287	0.2396	0.2541	0.2456	0.2388
300	0.4416	0.438	0.3583	0.4012	0.4101	0.3577

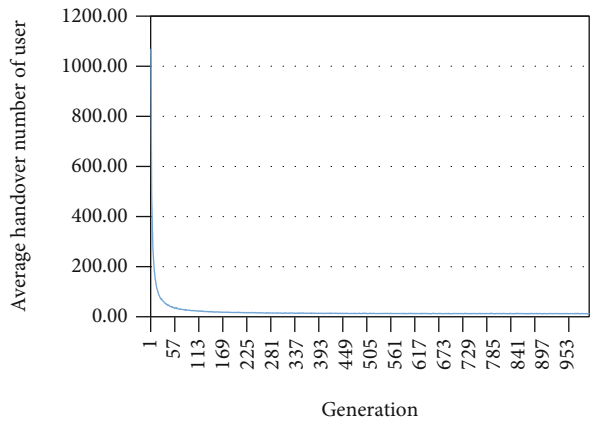


Figure 10: The convergence of LB-DDQN handover decision method.

5.3.5. The Convergence of the LB-DDQN Method. As Figure 10 shows, with the increases of generation number, the average handover number of users is convergent quickly. The number of mobile users is 100. When the number of generations is 127, the average handover number of users is 19.91. When the number of generations is 1000, the average handover number of a user is 11.32. Because of the random initialization for the Q-network, the initial value of average handover number for mobile users is very high. By the multiple iterations, experience replay, and small batch sampling methods, the weights of the main Q-network and the target Q-network are matured. The results of our method are also convergent rapidly.

6. Conclusions

The LB-DDQN handover decision method and the load balancing strategy are proposed in this paper. And the validation of our method is realized in the virtual simulation of the STIN. The designed load balancing strategy combines the load coefficient and load reward to assist the training of handover decision. The frequent handover and handover failure are optimized by the LB-DDQN handover decision method and load balancing strategy. The distributions of mobile users with different numbers are more balancing, and the number of handover failures is decreased. Furthermore, the LB-DDQN method adapts to the different conditions of user speeds, movement routes, and user numbers. Its adaptability and performance of handover decisions are good and low cost.

Data Availability

The data used to support the findings of this study are available from Dong-Fang Wu (at wudongfang@whu.edu.cn).

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 (No. 61772385).

References

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[2] G. Gódor, Z. Jakó, Á. Knapp, and S. Imre, "A survey of handover management in LTE-based multi-tier femtocell networks: requirements, challenges and solutions," *Computer Networks*, vol. 76, pp. 17–41, 2015.

[3] A. Stamou, N. Dimitriou, K. Kontovasilis, and S. Papavassiliou, "Autonomic handover management for heterogeneous networks in a future internet context: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3274–3297, 2019.

[4] N. P. Singh and B. Singh, "Vertical handoff decision in 4G wireless networks using multi attribute decision making approach," *Wireless Networks*, vol. 20, no. 5, pp. 1203–1211, 2014.

[5] S. Bhosale and R. Daruwala, "Multi-criteria vertical handoff decision algorithm using hierarchy modeling and additive weighting in an integrated WiFi/WiMAX/UMTS environment– a case study," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 1, pp. 35–57, 2014.

[6] M. Alhabo, L. Zhang, and N. Nawaz, "GRA-based handover for dense small cells heterogeneous networks," *IET Communications*, vol. 13, no. 13, pp. 1928–1935, 2019.

[7] B. R. Chandavarkar and R. M. R. Guddeti, "Simplified and improved multiple attributes alternate ranking method for vertical handover decision in heterogeneous wireless networks," *Computer Communications*, vol. 83, pp. 81–97, 2016.

[8] I. Bisio and A. Sciarrone, "Fast multiattribute network selection technique for vertical handover in heterogeneous emergency communication systems," *Wireless Communications and Mobile Computing*, vol. 2019, 17 pages, 2019.

[9] A. Çalhan and M. Cicioğlu, "Handover scheme for 5G small cell networks with non-orthogonal multiple access," *Computer Networks*, vol. 183, article 107601, 2020.

[10] E. R. Bastidas-Puga, Á. G. Andrade, G. Galaviz, and D. H. Covarrubias, "Handover based on a predictive approach of signal-to-interference-plus-noise ratio for heterogeneous cellular networks," *IET Communications*, vol. 13, no. 6, pp. 672–678, 2019.

[11] R. Arshad, H. ElSawy, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Velocity-aware handover management in two-Tier cellular networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1851–1867, 2017.

[12] Y. S. Hussein, B. M. Ali, M. F. A. Rasid, A. Sali, and A. M. Mansoor, "A novel cell-selection optimization handover for long-term evolution (LTE) macrocellusing fuzzy TOPSIS," *Computer Communications*, vol. 73, pp. 22–33, 2016.

[13] I. Kustiawan, C.-Y. Liu, and D. F. Hsu, "Vertical handoff decision using fuzzification and combinatorial fusion," *IEEE Communications Letters*, vol. 21, no. 9, pp. 2089–2092, 2017.

[14] X. Duan, J. Wei, D. Tian et al., "Adaptive handover decision inspired by biological mechanism in vehicle ad-hoc networks," *Computers, Materials & Continua*, vol. 61, no. 3, pp. 1117–1128, 2019.

[15] C.-C. Tseng, H.-C. Wang, K.-C. Ting, C.-C. Wang, and F.-C. Kuo, "Fast game-based handoff mechanism with load balancing for LTE/LTE-A heterogeneous networks," *Journal of Network and Computer Applications*, vol. 85, pp. 106–115, 2017.

[16] T. M. Duong and S. Kwon, "Vertical handover analysis for randomly deployed small cells in heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2282–2292, 2020.

[17] X. Xu, Z. Sun, X. Dai, T. Svensson, and X. Tao, "Modeling and analyzing the cross-tier handover in heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7859–7869, 2017.

[18] T. Coqueiro, J. Jailton, T. Carvalho, and R. Francês, "A fuzzy logic system for vertical handover and maximizing battery lifetime in heterogeneous wireless multimedia networks," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 1213724, 13 pages, 2019.

[19] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski et al., "Reinforcement learning for satellite communications: from LEO to deep space operations," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 70–75, 2019.

[20] A. M. Vegni and E. Natalizio, "A hybrid (N/M)CHO soft/hard vertical handover technique for heterogeneous wireless networks," *Ad Hoc Networks*, vol. 14, pp. 51–70, 2014.

[21] S. Zang, W. Bao, P. L. Yeoh, B. Vucetic, and Y. Li, "Managing vertical handovers in millimeter wave heterogeneous networks," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1629–1644, 2019.

[22] Y. Sun, G. Feng, S. Qin, Y.-C. Liang, and T.-S. P. Yum, "The SMART handoff policy for millimeter wave heterogeneous cellular networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1456–1468, 2018.

[23] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[24] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[25] A. D. D. M. Sana, E. C. Strinati, and A. Clemente, "Multi-agent deep reinforcement learning for distributed handover management in dense MmWave networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8976–8980, Barcelona, Spain, 2020.

[26] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4296–4307, 2018.

[27] J. Sun, Z. Qian, X. Wang, and X. Wang, "ES-DQN-based vertical handoff algorithm for heterogeneous wireless networks," *IEEE Wireless Communications Letters*, vol. 9, no. 8, pp. 1327–1330, 2020.

[28] T. Goyal and S. Kaushal, "Handover optimization scheme for LTE-advance networks based on AHP-TOPSIS and Q-learning," *Computer Communications*, vol. 133, pp. 67–76, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6646399, 13 pages
https://doi.org/10.1155/2021/6646399

Research Article

LRDC: Learning Resource Diffusion Based on Credibility for Computer-Supported Collaborative Learning

Peng Li,[1,2,3,4,5] Yuanru Cui,[5] Qian Liu,[5] Meirui Ren,[1,2,3,4,5] Longjiang Guo,[1,2,3,4,5] Hong Liu,[5] Lichen Zhang,[1,2,3,4,5] Xiaojun Wu,[1,2,3,4,5] Xiaoming Wang,[1,2,3,4,5] Ning An,[5] and Jinhu Sun[5]

[1]*Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China*
[2]*Key Laboratory of Intelligent Computing and Service Technology for Folk Song, Ministry of Culture and Tourism, Xi'an 710119, China*
[3]*Engineering Laboratory of Teaching Information Technology of Shaanxi Province, Xi'an 710119, China*
[4]*Xi'an Key Laboratory of Culture Tourism Resources Development and Utilization, Xi'an 710062, China*
[5]*School of Computer Science, Shaanxi Normal University, Xi'an 710119, China*

Correspondence should be addressed to Meirui Ren; meiruiren@snnu.edu.cn and Longjiang Guo; longjiangguo@snnu.edu.cn

Received 2 January 2021; Revised 6 September 2021; Accepted 23 October 2021; Published 30 November 2021

Academic Editor: Xiao Zhang

Computer-supported collaborative learning (CSCL) is a learning strategy that gathers students together on campus through mobile application software on intelligent handheld devices to carry out creative exploration learning activities and social interaction learning activities. Learning resource diffusion is a very important constituent part of CSCL mobile software. However, learners will receive or forward a large number of learning resources such that short video, images, or short audio which will increase the energy consumption of forwarding nodes and reduce the message delivery success rate. How to improve the message delivery success rate is an urgent problem to be solved. To solve the aforementioned problem, this paper mainly studies the diffusion of learning resources in campus opportunistic networks based on credibility for CSCL. In campus opportunistic networks, learners who participate in collaborative learning can obtain the desired learning resources through the distribution and sharing of learning resources. Learning resource diffusion depends on the credibility of learners who participate in collaborative learning. However, the existing classical algorithms do not take into account the credibility between learners. Firstly, the concept of credibility in campus opportunistic networks is proposed, and the calculation method of credibility is also presented. Next, the problem of node initialization starvation is solved in this paper. The node initialization starvation phase of collaborative learning is defined and resolved in campus opportunistic networks. Based on the information of familiarity and activity between nodes formed in the process of continuous interaction, a learning resource diffusion mechanism based on node credibility is proposed. Finally, the paper proposes a complete learning resource diffusion algorithm based on credibility for computer-supported collaborative learning (LRDC for short) to improve the delivery success rate of learning resources on the campus. Extensive simulation results show that the average message diffusion success rate of LDRC is higher than that of classical algorithms such as DirectDeliver, Epidemic, FirstContact, and SprayAndWait under the different transmission speed, buffer size, and initial energy, which is averagely improved by 46.83%, 44.43%, and 45.6%, respectively. The scores of LRDC in other aspects are also significantly better than these classical algorithms.

1. Introduction

Collaborative learning (CL) is a strategy that organizes some learners to form some groups or teams to carry out learning tasks [1]. In order to achieve the learning goal, all members of a collaborative learning group work together as a team to solve problems. The individual in a collaborative learning group can share learning resources with other members in the group, or with students outside the group. With the continuous innovation of educational ideology, the idea of

collaborative learning has been applied more and more widely, because it can better reflect the learner-centred educational ideology. In the process of group cooperation and interaction between learners, learners' participation and problem-solving skills are improved. Collaborative learning enhances the learning efficiency of individuals and the cohesion of the team in the collaborative group, which is also advantageous to enhancing learners' social skills. Therefore, CL is an effective way to improve learning efficiency.

Computer-supported collaborative learning (CSCL) is a learning strategy that gathers students together through mobile application software on intelligent handheld devices and promotes learners to carry out creative exploration activities and social interaction activities [2, 3]. Learning resource diffusion is a very important constituent part of CSCL mobile software. However, learners will receive or forward a large number of learning resources such that short video, images, or short audio which will increase the energy consumption of forwarding nodes and reduce the message delivery success rate. How to improve the message delivery success rate is an urgent problem to be solved. This is the motivation of this paper.

To solve the aforementioned problem, this paper mainly studies the diffusion of learning resources in campus opportunistic networks based on credibility for CSCL. In campus opportunistic networks, learners who participate in collaborative learning can obtain the desired learning resources through the distribution and sharing of learning resources. Learning resource diffusion depends on the credibility of learners who participate in collaborative learning. However, the existing classical algorithms do not take into account the credibility between learners.

First of all, this paper presents the definition of campus opportunistic networks. Opportunistic network [4, 5] is a kind of self-organizing network, which does not need a complete link between the source node and the destination node and uses the encounter opportunity brought by node movement to realize communication. The campus opportunistic network is a wireless self-organization network on campus, in which many learner nodes can participate in campus collaborative learning by using handheld smart devices. Every learner is a learner node participating in communication, and the communication manner depends on learner's community and the opportunistic networks formed by the information of the learners in the community.

Next, the concept of credibility in campus opportunistic network is proposed, and the calculation method of credibility is also presented in this paper. In campus opportunistic networks, learning resource diffusion is a very important constituent part for CSCL mobile software. In campus opportunistic networks, the learners who participate in collaborative learning can obtain the desired learning resources through the distribution and sharing of learning resources. Learning resource diffusion depends on credibility of learners who participate in collaborative learning. The diffusion of campus learning resources depends on the credibility between nodes. Because there is no credibility, resource diffusion between nodes is meaningless. However, the existing classical algorithms such as Epidemic [6], DirectDeliver [7], FirstContact [8], and SprayAndWait [9] do not take into account the credibility between learners. These algorithms are not suitable for resource diffusion on campus opportunistic network. This paper studies the diffusion of learning resources in campus opportunistic networks based on credibility for CSCL.

Then, this paper solves the problem of the node initialization starvation on campus opportunistic networks. Campus opportunistic networks have the characteristics of high node density, complex social relations among individuals and communities, and the complicated links among nodes. So, it is particularly difficult to select the suitable node to forward learning resources. The selection of the suitable forwarding node depends on community to which the learner belongs. The community difference between nodes is a key factor for selecting suitable forwarding nodes [10]. However, community difference among nodes is not significant in the early stage forming collaborative learning community. Therefore, it is difficult to select forwarding nodes in this stage. In this paper, in the early stage of community formation, the selection of forwarding nodes is called as the node initialization starvation stage of collaborative learning in campus opportunistic networks. In social networks, the current mainstream solutions of node initialization starvation problem focus on the problems of information overload, accuracy, and efficiency of recommendation. However, node initialization starvation problem of collaborative learning in opportunistic networks is quite different from that of mainstream social networks, because in the general social networks [11], there will be no message forwarding delay which caused by insufficient contact between mobile nodes. Therefore, the solution of node initialization starvation problem in the general social networks cannot be directly applied to campus opportunistic networks.

Finally, for node initialization starvation problem, this paper defines the node initialization starvation stage of collaborative learning in campus opportunistic networks and proposes LRDC algorithm. In the early stage of collaborative learning in campus opportunistic networks, the effective information collected by mobile learner nodes is less, and the community formed is relatively unstable. In order to obtain the effective information of nodes in the community as soon as possible, complete the information transmission between nodes in the node initialization starvation stage, and improve the transmission efficiency of learner nodes, this paper defines the node initialization starvation stage of collaborative learning in campus opportunistic networks.

In this paper, activity and familiarity are comprehensively considered as credibility to select forwarding nodes. According to the rule of node movement, the process of learning resource diffusion is divided into node initialization starvation stage and community operation stage. A novel learning resources diffusion algorithm based on credibility named LRDC is proposed in this paper. For collaborative learning in campus opportunistic networks, the contributions of this paper are given as follows:

(i) First of all, this paper defines the concept of campus opportunistic network and proposes the concept of

credibility in campus opportunistic network and gives the calculation method of credibility, which is not found in other papers. According to the continuous interaction information between the learner nodes, the familiarity and activity of the nodes are calculated, the credibility of the nodes is accurately evaluated, and the diffusion of learning resources is completed efficiently based on the credibility

(ii) Next, this paper effectively solves the problem of node initialization starvation. The learner node can quickly obtain the effective information of the node initialization starvation stage in the learning community, complete the information transmission between the nodes in the node initialization starvation stage, and improve the transmission efficiency

(iii) Finally, this paper proposes a complete LRDC algorithm to diffuse learning resources on campus opportunistic networks and improve the delivery success rate of learning resources on the campus. An extensive simulation results show that the average message diffusion success rate of LDRC is higher than that of classic algorithms such as DirectDeliver, Epidemic, FirstContact, and SprayAndWait under the different transmission speed, buffer size, and initial energy, which is averagely improved by 46.83%, 44.43%, and 45.6%, respectively. The scores of LRDC in other aspects is also significantly better than these algorithms

The rest of this paper is organized as follows. The second section introduces related work. The third section describes the starvation phase division of learner node initialization. The fourth section introduces modelling process of credibility degree for campus collaborative learning nodes. The fifth section describes the diffusion mechanism of learning resources in the collaborative learning environment of campus opportunity network. The sixth section carries on the simulation and the result analysis. At last, the seventh section summarizes the full paper.

2. Related Work

In order to solve the information transmission between the nodes in the node initialization starvation stage, researchers propose different routing algorithms for different situations. In social networks, initialization stage refers to the use of a large number of newly created services and these services of no usage records [12], the node initialization starvation problem can be divided into new project initialization starvation problem and new user initialization starvation problem [13]. In order to solve the node initialization starvation problem in social networks, Huang et al. proposed and implemented the prediction of the potential collaboration relationship between users and the relevant recommendation mechanism based on the model of service relationship [14]; Zhang et al. parsed user's demand information based on the LDA (latent Dirichlet allocation) model and keyword information to improve accuracy of the recommendation [15]. In social networks, the above mainstream solutions of the node initialization starvation problem are mainly aimed at providing customers with product information and suggestions and dealing with the problems of information overload, accuracy, and efficiency of recommendation and do not consider the node initialization starvation problem in collaborative learning opportunistic networks and the distribution and forwarding of learning resources with social intimacy and node influence. In campus opportunistic networks, the node initialization starvation problem refers to that in the initial stage of opportunistic network; due to the insufficient information of link connection and the delay in obtaining the information of each node, the interaction between nodes and data diffusion are blind, and the message delivery rate is low; at the same time, the transmission opportunity is wasted. Therefore, the above schemes are not suitable for collaborative learning opportunistic networks. This paper considers the node initialization starvation problem of campus collaborative learning opportunistic networks, make full use of contact information between nodes and the characteristics of nodes, and promote the efficient and rapid transmission of node information in the campus collaborative environment.

In the opportunistic network, relevant scholars have carried out a series of explorations on familiarity and activity of nodes. For example, Wang et al. [16] calculated the node activity according to the area covered by moving trajectory of the node and the average dwell time of the node. Feng et al. [17] proposed a delayed back off algorithm based on the active state of neighbour nodes; in the algorithm, the delayed back off window is adaptively adjusted according to the active number of neighbour nodes. Rao et al. [18] considers the link similarity of nodes according to the node link condition in k-hop. The LFM algorithm proposed by Lancichinetti et al. [19] is based on the consideration of the similarity between the feature of nodes in contact with each other. However, these algorithms are not suitable as the diffusion mechanism of collaborative learning resources in campus opportunistic networks.

Among the classical algorithms in opportunistic networks, Amin and David [6] proposed a flooding algorithm named Epidemic similar to the spread of infectious diseases. When two nodes meet, they exchange packets carried by each other. This algorithm has large network load and poor scalability. Shah et al. [7] proposed an algorithm named DirectDeliver that the source node forwards the data only when it meets the destination node. This algorithm has poor performance in delivery rate and delay. Zhao et al. [8] proposed FirstContact algorithm that the source node forwards the data to the meeting node, and then, these mobile nodes forwarded the data to the destination node. Spyropoulos et al. [9] proposed SprayAndWait algorithm to limit the number of data backup. When the number of data backup in the network reaches a certain value, the data will stop spreading and adopt the strategy of direct waiting. The above algorithms have defects in message delivery rate and delivery delay, and they all do not consider credibility between learners in campus opportunistic networks. Therefore, this paper proposes LRDC algorithm to solve the problem of learning resource diffusion based on creativity for computer-supported collaborative learning, and after

Table 1: Node initialization information table.

Variable name	Meaning
NS_{ij}	Number of encounters between node i and node j
Sig_i	The signal value of the node i
Init $i\left\{NS_{ij_1},,,NS_{ij_p}\right\}\left\{0 \le j_1 \le j_p \le n\right\}$	Initial queue
Q	Community modularity
m	Total number of edges in the network
d_j	Degree of node j
A	Initial matrix

verification in extensive simulation results, LRDC is better than these classical algorithms.

3. The Learner Node Initialization Starvation Phase

In the mobile social network, a single learner node is used as a unit for message transmission. Therefore, in the collaborative learning community based on mobile social networks, the running process of the network can be divided into two phases such as the node initialization starvation phase and the normal operation phase according to the communication situation of a single node. In the campus, the learner nodes are denser, and every learner's daily schedule has more obvious regularity and shows more obvious characteristics of group activities, so the community between learner nodes is more obvious.

Campus collaborative learning community is a virtual learning organization in which students participate in groups to maximize the acquisition results under a certain incentive mechanism. In the learning community, students cooperate each other based on their own will and the common goal.

As previously mentioned, in the early stage of collaborative learning in campus opportunistic networks, the effective information collected by mobile learner nodes is less, and the formed community is relatively unstable, so this paper will divide the running process of campus opportunistic networks into two phases: the node initialization starvation phase and the normal operation phase.

The node initialization starvation phase is the phase in which the community of the node is in an unstable state. When the community of a learner node is in a stable state, it can be considered that the node initialization starvation phase will end.

Adjacency matrix is the most suitable data structure to express social network relation, NS represents the set of the number of encounters between two nodes, the element NS_{ij} indicates the number of encounters between node i and node j. For any node i, set the initial value of its signal $Sig_i = n - 1$ (n is the upper limit of the number of students in the campus collaboration network), initial queue, initial matrix, and other parameters are maintained (as shown in Table 1).

Next, we show how to determine whether the community in which the node is located is stable, that is, the node initialization starvation phase ends. Firstly, we introduce the initialization adjacency matrix to represent the encounter situation between nodes in the starvation initialization phase. The formation process of the initialization adjacency matrix is as follows:

Case 1. When node p encounters to node q and the signal value of node p (p can be any node) and the signal value of node q are not greater than 0, the signal values of p and q will not be changed.

Case 2. When the signal value of node p is greater than 0 (p can be any node) and node p encounters to node q whose signal value is greater than 0, the signal value of node p and node q are subtracted by one, and the signal value of node q in node p's signal queue is increased by one, simultaneously, and the signal value of node p in node q's signal queue is also increased by one.

Case 3. When the signal value of node p is not greater than 0 (p can be any node), and node p encounters to node q whose signal value is greater than 0:

(a) If the queue of node p does not contain node q, the signal value of node p will be kept original, and the node q will be added to the queue of node p, and the signal value of the node q in the node p's queue will be added by one. At the same time, node p is added to the queue of node q, the signal value of node q is subtracted by two, and the signal value of node p in node q's queue is added by one

(b) If the initial queue of node p contains node q, the signal value of node q in node p's signal queue is increased by one, and the signal value of node p in node q's signal queue is added by one, and the signal value of node q itself is subtracted by two

Case 4. When the signal value of node p is greater than 0 (p can be any node), and node p encounters to node q whose signal value is not greater than 0, node p and node q do nothing.

When the signal values of all nodes are less than or equal to 0, the initialization phase is completed. At this time, we need to further judge through the community modularity.

In this section, we use modularity which is a common method to measure the stability of network community to detect the initial communities [20, 21]. The definition of community modularity Q is as follows:

$$Q = \frac{1}{2m}\sum_{ij}\left(NS_{ij} - \frac{d_i d_j}{2m}\right)\varepsilon(c_i, c_j), \tag{1}$$

where m is the total number of edges in the network, d_i and d_j, respectively, represent the degree of node i and node j. NS_{ij} represents the number of encounters between node i and node j; if there is no encounter between node i and node j, the value of NS_{ij} is 0; otherwise, it is equal to the number of encounters. c_i represents the community where node i is located, and c_j represents the community where node j is located. If node i and node j are in the same community, the value of $\varepsilon(c_i, c_j)$ is 1; otherwise, $\varepsilon(c_i, c_j)$ is 0.

In this scenario, c_i and c_j are unknown, so the value of $\varepsilon(c_i, c_j)$ cannot be confirmed, but it can be deduced according to the number of encounters between node i and node j. If the number of encounters between node i and node j is greater than the average number of encounters between node i and all other nodes in the community, it indicates that node i and node j are in the same community.

Take node i as an example, and the number of encounters between node i and node j is 5, and the average number of encounters between node i and all initial nodes is 2; it is obvious that node i and node j are in the same community.

The value range of Q is [0, 1], but in the actual network, the value of Q is usually between (0.3, 0.7) [22, 23]. So, in this paper, we set that when the Q value is greater than 0.3, the node starvation phase ends, and the network is in a relatively stable community.

This paper simulates and calculates the value of modularity in campus collaborative learning community, and through calculation, it is concluded that the Q is greater than 0.3. At this point, it is determined that the community is in a stable state and obtained the following initialization matrix NS which represents the community network structure.

4. Model for Credibility between Nodes

In this section, we introduce the concept of credibility between nodes. This concept serves as a metric for selecting the next forwarding node in the normal operation phase. In this section, there basic variables are shown in Table 2.

4.1. Familiarity

4.1.1. Matching Matrix between Nodes. Suppose that NS represents the community network structure. A specific community network structure is given as follows.

Table 2: Variable name table.

Variable name	Meaning
C_{ij}	The credibility of node j to node i
AD_j	Activity of node j
FD_{ij}	Familiarity between node i and node j
S_i	The sum of weights of all edges connected to node i
WE_{ij}	The edge weight between nodes i and j
i_Asb	The direct node set of node i

$$NS = \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \\ i_9 \\ i_{10} \\ i_{11} \\ i_{12} \\ i_{13} \\ i_{14} \end{array} \begin{array}{c} \begin{array}{cccccccccccccc} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 & i_9 & i_{10} & i_{11} & i_{12} & i_{13} & i_{14} \end{array} \\ \begin{bmatrix} 0 & 4 & 2 & 1 & 3 & 4 & 3 & 2 & 0 & 0 & 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 4 & 3 & 0 & 0 \\ 3 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 0 & 4 & 0 & 0 & 0 \\ 3 & 2 & 0 & 2 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 4 & 2 & 4 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}. \tag{2}$$

Matrix A_{i_1} represents the adjacency relationship between nodes in the community with the node i_1 as the center node. Suppose that A_{i_1} based on the above NS is given as follows.

$$A_{i_1} = \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \\ i_{11} \end{array} \begin{array}{c} \begin{array}{ccccccccc} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 & i_{11} \end{array} \\ \begin{bmatrix} 0 & 4 & 2 & 1 & 3 & 4 & 3 & 2 & 3 \\ 4 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 3 & 0 & 2 & 0 & 2 & 0 & 4 \\ 3 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 2 \\ 4 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 4 \\ 3 & 2 & 0 & 2 & 0 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 2 \\ 3 & 0 & 0 & 4 & 2 & 4 & 0 & 2 & 0 \end{bmatrix} \end{array}$$

Taking the adjacency matrix of aforementioned community network structure NS as an example, suppose that node i_1 carries messages, and nodes i_1 and i_6 are candidate transmission nodes. The matching matrix for node i_6 to node i_1 is calculated as follows: the number of rows and columns of matrix $A_{i_6\&i_1}$ is the same as that of A_{i_1}, and if there is a nonzero element in matrix A_{i_1} and its corresponding position in matrix A_{i_6} is also a nonzero element, then the value of the element at that position in matrix $A_{i_6\&i_1}$ is the same as that

of A_{i_1}; otherwise, the value is 0 in matrix $A_{i_6\&i_1}$. In the same way as above, based on $A_{i_6\&i_1}$ and A_{i_7}, the matching matrix $A_{i_7\&(i_6\&i_1)}$ can be obtained. For example, the matching matrix $A_{i_6\&i_1}$ and $A_{i_7\&(i_6\&i_1)}$ are as follows:

$$A_{i_6\&i_1} = \begin{bmatrix} 0&0&0&0&0&4&3&2&3\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 4&0&0&0&0&0&1&2&4\\ 3&0&0&0&0&1&0&1&0\\ 2&0&0&0&0&2&1&0&2\\ 3&0&0&0&0&4&0&2&0 \end{bmatrix},$$

$$A_{i_7\&(i_6\&i_1)} = \begin{bmatrix} 0&0&0&0&0&4&3&2&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 4&0&0&0&0&0&1&2&0\\ 3&0&0&0&0&1&0&0&0\\ 2&0&0&0&0&2&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{bmatrix}. \tag{3}$$

4.1.2. Familiarity between Nodes. In this section, this paper calculates the social familiarity between nodes through the situation of nodes' common neighbors. The familiarity between node i and node j is to be calculated as follows:

$$FD_{ij} = SN_{ij} \cdot SC_{ij}, \tag{4}$$

where FD_{ij} is the familiarity between node i and node j and SN_{ij} is the proportion of the common neighbors of nodes i and j to all the nodes contacted by nodes i and j, and SC_{ij} is the link contribution of the common neighbors of nodes i and node j to nodes i and j. WE_{ij} is the edge weight between node i and node j, and S_i is the sum of the weights of all edges connected to node i. The edge strength ES_{ij} is defined as the ratio of WE_{ij} to the sum of S_i and S_j. The calculations of SN_{ij}, ES_{ij}, and SC_{ij} are shown in formulas (5), (6), and (7), respectively.

$$SN_{ij} = \frac{|i\_Asb \cap j\_Asb|}{|i\_Asb \cup j\_Asb|}, \tag{5}$$

$$ES_{\mathrm{ij}} = \frac{\mathrm{WE}_{ij}}{S_{\mathrm{i}} + S_j}, \tag{6}$$

$$SC_{ij} = \sum_{c \in i\_Asb \cap j\_Asb} \left\| A_{c\&(i\&j)} \right\|_F^2 \cdot ES_{ij}. \tag{7}$$

In formula (7), c is any node in the set of common neighbor nodes of nodes i and j, i_Asb is the direct node set of node i, $A_{c\&(i\&j)}$ is the matching matrix of c for nodes i and j, and $\|A_{c\&(i\&j)}\|_F^2$ represents the F-norm of the matrix $A_{c\&(i\&j)}$. It can be considered that the larger the SC_{ij} is, the higher the contribution of the shared neighbor nodes to the links of nodes i and j. Let us take adjacency matrix NS in Section 3 as an example to calculate the familiarity between node i_1 and node i_6.

$$ES_{i_1 i_6} = \frac{4}{(4+2+1+3+4+3+2+3)+(4+1+2+2+4)} = \frac{4}{35},$$

$$SC_{i_1,i_6} = \left(\left\| A_{i_2\&(i_6\&i_1)} \right\|_F^2 + \left\| A_{i_8\&(i_6\&i_1)} \right\|_F^2 + \left\| A_{i_{11}\&(i_6\&i_1)} \right\|_F^2 \right) \cdot ES_{i_1 i_6}$$

$$= \left[\left\| \begin{bmatrix} 0&0&0&0&0&4&3&2&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 4&0&0&0&0&0&1&2&0\\ 3&0&0&0&0&1&0&0&0\\ 2&0&0&0&0&2&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{bmatrix} \right\|_F^2 + \left\| \begin{bmatrix} 0&0&0&0&0&4&3&2&3\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 4&0&0&0&0&0&1&2&4\\ 3&0&0&0&0&1&0&0&0\\ 2&0&0&0&0&2&0&0&2\\ 3&0&0&0&0&4&0&2&0 \end{bmatrix} \right\|_F^2 + \left\| \begin{bmatrix} 0&0&0&0&0&4&0&2&3\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 4&0&0&0&0&0&0&2&4\\ 0&0&0&0&0&0&0&0&0\\ 2&0&0&0&0&2&0&0&2\\ 3&0&0&0&0&4&0&2&0 \end{bmatrix} \right\|_F^2 \right] \times \frac{4}{35} = 32.46, \tag{8}$$

$$SN_{i_1,i_6} = \frac{4}{22+11} = \frac{4}{33},$$

$$FD_{i_1,i_6} = \frac{4}{33} \times 32.46 = 3.93.$$

4.2. Activity of Node. In the early stage of learning community in campus opportunistic networks, because any node does not contain or only contains less communication characteristic information of other nodes, when any node needs to send messages, it cannot select the forwarding node correctly according to the node information table it maintains, which will cause a lot of contact opportunity waste, message redundancy, and the increase of message transmission delay.

When the node in initialization starvation phase performs message transmission, it will select the next hop node by comprehensive consideration of its activity and familiarity.

When calculating the activity of node i (marked as AD_i), the value of AD_i can be divided into direct contact degree AD_dir_i and indirect contact degree AD_rel_i as shown in formula (9); AD_dir_i represents the ability that node i contacts directly other nodes, and AD_rel_i reflects the ability that node i to contact indirectly other nodes through its direct neighbors, as shown in the formulas (10) and (11), respectively. In formula (11), $NS_{ij(c)}$ represents the number of times that node j contacts the common neighbors of node i and node j, and it can be calculated by formula (12).

$$AD_i = AD\_dir_i + AD\_rel_i, \tag{9}$$

$$AD\_dir_i = \sum_{j \in i\_Asb} NS_{ij}, \tag{10}$$

$$AD\_rel_i = \sum_{j \in i\_Asb} \frac{AD\_dir_j - NS_{ij(c)} - NS_{ij}}{AD\_dir_j} \times NS_{ij}, \tag{11}$$

$$NS_{ij(c)} = \sum_{k \in (i\_Asb \cap j\_Asb)} NS_{jk}. \tag{12}$$

4.3. Credibility of Nodes. In order to solve the problem of finding learning resources for learners in collaborative learning environment, this paper proposes a learning resource diffusion strategy based on credibility for computer-supported collaborative learning. According to common sense, a node that you are familiar with and active, you will believe that it has the ability to spread messages. So, a learner node wants to forward its learning resources as soon as possible; it needs to find a node that is familiar with it and highly active. Therefore, the paper defines the credibility as the weighted sum of familiarity and activity (as shown in formula (13)). The credibility of nodes i to j is denoted C_{ij}, in particular, C_{ij} and C_{ji} are not necessarily equal. AD_j is the activity of node j, the familiarity between nodes i and j is represented by FD_{ij}, and θ represents the weight coefficient.

$$C_{ij} = \theta \cdot AD_j + (1 - \theta) \cdot FD_{ij}. \tag{13}$$

This paper uses formula (14) to normalize the value of C_{ij}, where max and min represent, respectively, the maximum and minimum values of credibility.

$$X^* = \frac{x - \min}{\max - \min}. \tag{14}$$

5. Learning Resource Diffusion Algorithm Based on Credibility

In the context of collaborative learning on campus, the movement of learners' nodes has strong regularity and community. Based on the initialization process of node starvation phase proposed in Section 3 and the calculation method of credibility proposed in Section 4, this section proposes LRDC (learning resource diffusion based on credibility) algorithm and discusses the learning resource diffusion strategy in cooperative learning community as follows.

For any node i, establish direct contact node linked list $\psi_{i\_all} = (i, c_1, c_2, \cdots, c_m, d_1, d_2, \cdots, d_n)$, community node linked list $\psi_{i\_com} = (i, c_1, c_2, \cdots, c_m)$, and noncommunity node linked list $\psi_{i\_dir} = (i, d_1, d_2, \cdots, d_n)$, respectively. In ψ_{i_all}, $c_i (i = 1, 2, \cdots, m)$ indicates the node whose contact times with node i is greater than the average contact number between node i and all nodes, and $c_i (i = 1, 2, \cdots, m)$ will be added to the community node linked list ψ_{i_com}; $d_j (j = 1, 2, \cdots, n)$ indicates the node whose contact times with node i is not greater than the average contact number between node i and all nodes, and $d_j (j = 1, 2, \cdots, n)$ will be added to the community node linked list ψ_{i_dir}.

The first element in the above each linked list is the central node i, and the other elements are sorted according to the credibility of the node i to them from large to small. When the credibility is the same, they are sorted according to the times of contact with node i. When any two nodes contact each other, they update their node linked lists according to the contact information of each other. The contact linked list representation of node i is shown in Figure 1.

Region 1 is the community node linked list of node i, region 2 is the noncommunity node linked list $\psi_{i\_dir} = (i, d_1, d_2, \cdots, d_n)$ of node i and the community node linked list of each element except node i in ψ_{i_com}, and region 3 is the noncommunity node linked list of each element except node i in ψ_{i_com}.

According to the elements in the linked lists of region 2 and region 3, the matrices A_{id} and A_{ic} are established, respectively, as follows:

$$A_{id} = \begin{bmatrix} d_1 & c_1c_1 & c_2c_1 & \cdots & c_mc_1 \\ d_2 & c_1c_2 & c_2c_2 & \cdots & c_mc_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_n & c_1c_m & c_2c_m & \cdots & c_mc_m \end{bmatrix},$$
$$A_{ic} = \begin{bmatrix} c_1d_1 & c_2d_1 & \cdots & c_md_1 \\ c_1d_2 & c_2d_2 & \cdots & c_md_2 \\ \cdots & \cdots & \cdots & \cdots \\ c_1d_n & c_2d_n & \cdots & c_md_n \end{bmatrix}. \tag{15}$$

Taking node i as an example, the priority determination strategy for selecting forwarding nodes is as follows: the priority of nodes in the linked list ψ_{i_com} is the highest, the

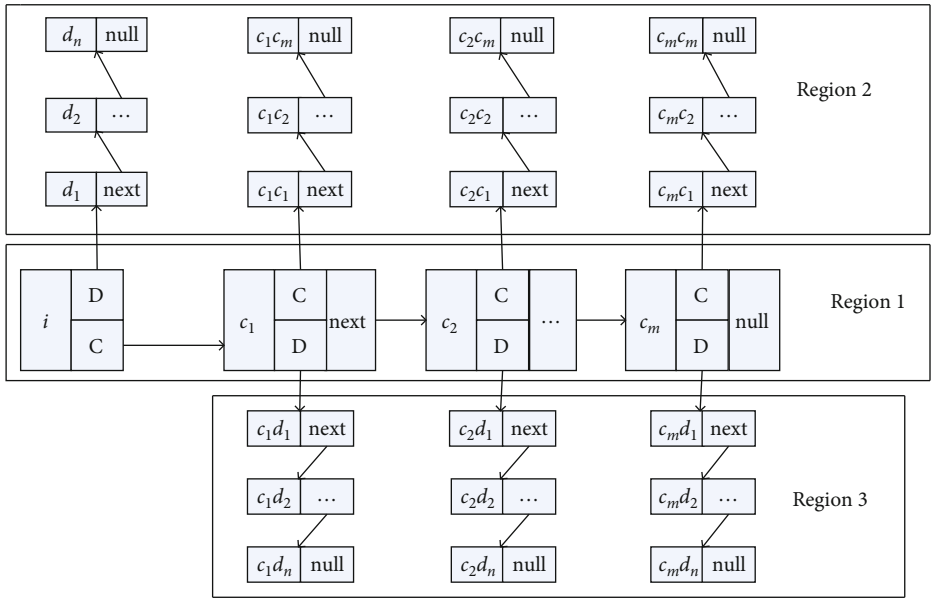


Figure 1: Node contact linked list diagram.

priority of nodes in A_{id} matrix is the second, and the priority of nodes in A_{ic} matrix is the lowest.

Suppose that node i is the sending node and j is the destination node. The routing and forwarding scheme is described as below:

(1) If there is a destination node j in the community node linked list of node i, the node i's message is sent directly to the destination node j; otherwise, the matrix A_{ic} and A_{id} are traversed according to the priority order, respectively, and if the node j exists in matrix A_{ic} or A_{id}, the node i's message is forwarded to the destination node j by the corresponding relay nodes

(2) If the situation in (1) does not occur, the message is not forwarded

The LRDC algorithm is described in Figure 2.

6. Simulation and Result Analysis

6.1. Simulation Setup. In order to evaluate the performance of the algorithm described in this paper, ONE (Opportunistic Network Environment) simulator is used for experimental simulation. In this paper, the real dataset CRAWDAD dataset Cambridge/haggle [24] is used for simulation experiment, and the main experimental parameters are set as shown in Table 3.

6.2. The Evaluation on Message Transmission. This section presents a set of experiments comparing the performance of LRDC, DirectDeliver, Epidemic, FirstContact, and SprayAndWait and considering the effect of simulation time on the message delivery success rate, the average residual energy of nodes, and the average hops of message transmission.

This section has two goals. The first goal is to quantify the effect of simulation time on the message delivery success rate; the second goal is to quantify the average residual energy of nodes and the average hops of message transmission after 90000 s simulation time. Set other main parameters of simulation environment to $initialEnergy = 50000$, $bufferSize = 100$ MB, $transmitSpeed = 150$ KB/s, and $msgttl = 900$ min; the simulation results are given in Tables 4 and 5.

For the first goal, with the increase of simulation time, the success rate of message delivery of five routes is increasing continuously. And in the same simulation time period, LRDC's message delivery success rate is the highest among the five routing algorithms. It can be seen that LRDC can consume less energy and deliver messages to the destination node at a faster speed without causing message redundancy when the community of the node is not fully mature, and its delivery time can approach the theoretical limit.

For the second goal, this section has three subgoals. For the first subgoal, the higher the average residual energy, it indicates that the performance of routing algorithm is better. It can be seen from Table 5 that the average residual energy of DirectDeliver is the highest, but the routing algorithm is not suitable for the campus collaborative learning situation based on opportunistic network. The average residual energy of SprayAndWait and LRDC is also high, but the average delay of message transmission of SprayAndWait is poor. For the second subgoal, the shorter the average delay time of message transmission, it indicates that the performance of routing algorithm is better. It can be seen from Table 5 that the average delay time of message transmission of Epidemic is the shortest and that of LRDC is also shorter, but the average residual energy of nodes of LRDC is much higher than that of Epidemic. For the third subgoal, the lower the average hops of message transmission, the lower the network overhead; it indicates that the performance of routing algorithm is better. The average hops of message

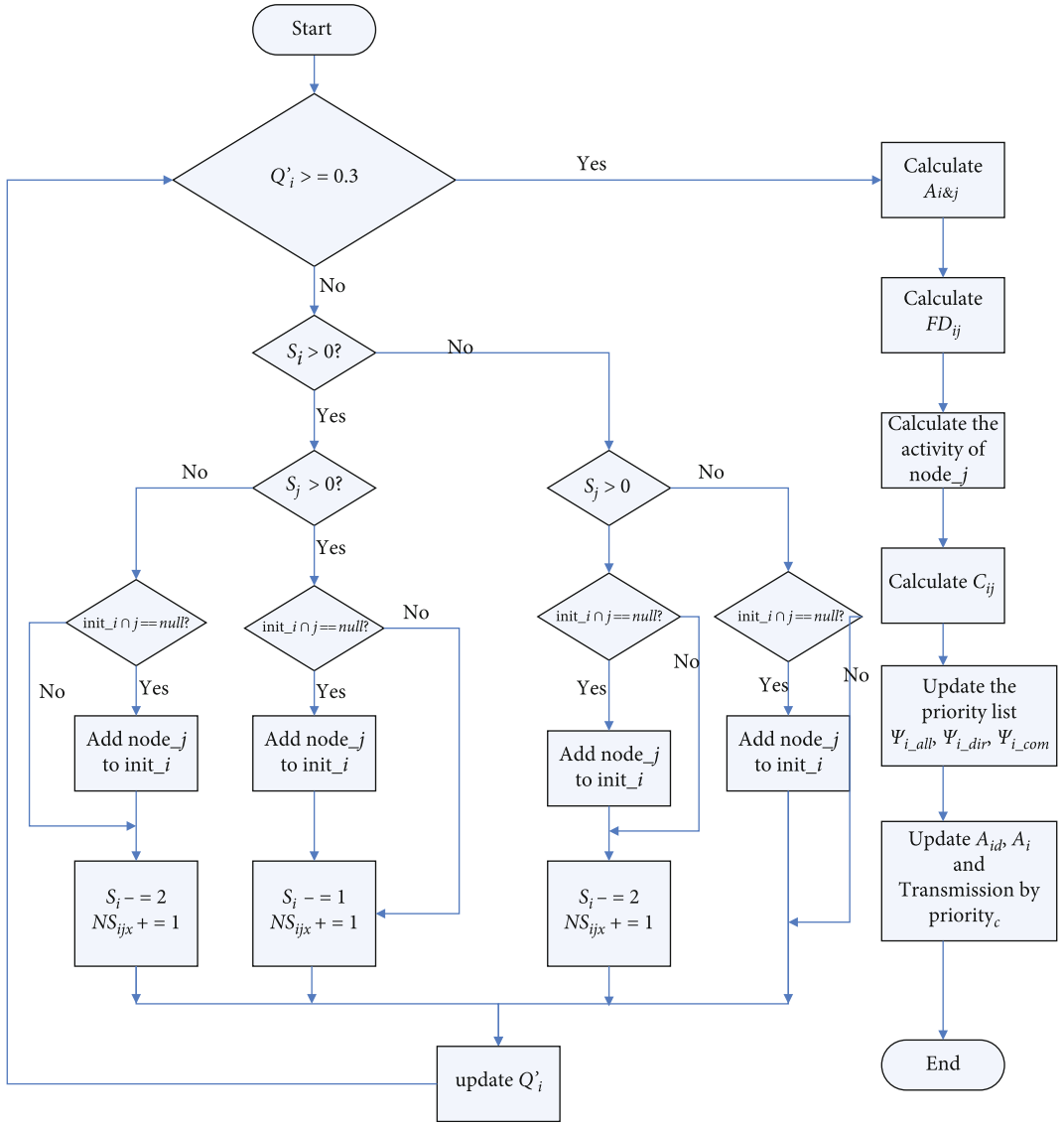


Figure 2: The flow chart of node initialization and learning resource diffusion in LRDC.

Table 3: The simulation parameter settings.

Parameter type	Value
Data set	CRAWDAD dataset Cambridge/haggle
Simulation duration/s	92000
Transfer file size/kB	50 k, 5000 k
Message generation interval/s	300, 5000
Number of nodes	98
Number of message copies from	20

transmission of DirectDeliver is the lowest, but the routing algorithm is not suitable for the campus collaborative learning situation based on opportunistic network. The average residual energy of LRDC and SprayAndWait is also low, but the average delay of message transmission of SprayAndWait is poor.

It can be seen from Tables 4 and 5 that the performance of LRDC, the average delay of message transmission, and the average hop number of message transmission are the best, so the performance of LRDC is the best.

This paper uses data normalization to process Table 5 and get the normalized score table in Table 6. Data

Table 4: The effect of simulation time on message delivery success rate.

Algorithm/simulation time (ten thousand seconds)	1	2	3	4	5	6	7	8	9	Average success rate	Relative promotion ratio (%)
LRDC	0.07	0.1	0.2	0.27	0.29	0.53	0.62	0.74	0.94	0.42	0
DirectDeliver	0	0	0	0.12	0.18	0.32	0.35	0.35	0.35	0.15	176
Epidemic	0.06	0.09	0.18	0.27	0.29	0.53	0.62	0.74	0.94	0.35	19
FirstContact	0	0	0	0.12	0.24	0.35	0.5	0.62	0.62	0.27	53
SprayAndWait	0	0.06	0.09	0.18	0.26	0.47	0.59	0.68	0.68	0.33	25
Average relative promotion ratio(%)											68.25

Table 5: The evaluation on message transmission.

Algorithm	The average residual energy	The average delay	The average hops
LRDC	40970.775	8864.43	5.75
DirectDeliver	45288.5	15562.93	1
Epidemic	19691.025	8816.55	6.22
FirstContact	40186	22133.09	28.29
SprayAndWait	42298.025	14085.71	4.52

normalization is as follows: because the higher the average residual energy, it indicates that the performance of routing algorithm is better. This paper uses formula (16) to deal with the data of the average residual. Because the shorter the average delay time of message transmission, it indicates that the performance of routing algorithm is better. Because the lower the average hops of message transmission, the lower the network overhead, it indicates that the performance of routing algorithm is better. This paper uses formula (17) to deal with the data of the average delay and the average hops. This section uses max and min to represent the maximum and minimum values of five kinds of routing data, respectively. For the calculation of total score, we use formula (18) to deal with it. The higher the total score, the better the routing algorithm performance. It can be seen from Table 6 that the LRDC has the highest total score, so the performance of LRDC is the best.

$$X^* = \frac{x - \min}{\max - \min}, \tag{16}$$

$$X = 1 - \frac{x - \min}{\max - \min}, \tag{17}$$

$$\text{Total score} = \text{TNARE} + \text{TNAD} + \text{TNAH}, \tag{18}$$

6.3. The Effect of Message Lifetime on LRDC Message Delivery Success Rate. When other main parameters of the simulation environment are set to $initialEnergy = 50000$, $buffer Size = 100$ MB, and $transmitSpeed = 150$ KB/s, the effect of msgTtl on LRDC is tested. The simulation results are as follows:

This section presents a set of experiments comparing the performance of LRDC, DirectDeliver, Epidemic, FirstContact, and SprayAndWait and considering the effect of message lifetime on message delivery success rate.

This paper uses data normalization to process Table 7 and uses formula (16) to deal with the data of the RECMSR (ratio of energy consumption to message delivery success rate). And define the following formula:

$$\text{RECMSR} = \frac{\text{the average success delivery rate}}{\text{5000-the residual enery}}. \tag{19}$$

As can be seen from Table 7, when TTL is less than 400 min, the impact on each routing algorithm is greater. When TTL is greater than 400 min, the performance of each routing algorithm is gradually stable. Combining with Table 5, it can be seen that the message transmission delay of LRDC and Epidemic is the lowest. Since the message transmission delay of the FirstContact is the highest, the change of TTL also has a great influence on it. The change of TTL also has a great impact on FirstContact, and has little impact on LRDC and Epidemic. The higher the normalized RECMSR, the better the routing algorithm performance. Among them, the normalized RECMSR of LRDC is the highest, so the performance of LRDC is the best.

6.4. The Effect of Transmission Speed on LRDC Message Delivery Success Rate. Set other main parameters of simulation environment to $initialEnergy = 50000$, $buffer Size = 100$ MB, and $msgTtl = 900$ min, the effect of transmitSpeed on LRDC was tested. The simulation results are as follows.

This section presents a set of experiments comparing the performance of LRDC, DirectDeliver, Epidemic, FirstContact, and SprayAndWait and considering the influence of transmission speed on message delivery success rate.

It can be seen from Table 8 that when the *transmitSpeed* is less than 50 kb/s, the impact on LRDC, DirectDeliver, Epidemic, FirstContact, and SprayAndWait is greater than that on DirectDeliver. When the *transmitSpeed* is greater than 50 kb/s, the performance of each routing algorithm gradually stabilizes. It can be seen from Table 5 that the DirectDeliver has less hops for message delivery. It directly transmits the message to the destination node. Therefore, when the *transmitSpeed* is limited, the effect on the DirectDeliver is low, but the overall performance of the routing algorithm is poor.

$$\text{Relative promotion ratio} = \frac{\text{the difference average success delivery rate between LRDC and } R}{\text{the average success delivery rate of } R}. \tag{20}$$

Table 6: The normalized score table.

Algorithm/scoring project	The normalized average residual energy (TNARE)	The normalized average delay (TNAD)	The normalized average hops (TNAH)	Total score
LRDC	0.8313	0.9964	0.8259	2.6536
DirectDeliver	1	0.4934	1	2.4934
Epidemic	0	1	0.8087	1.8087
FirstContact	0.8007	0	0	0.8007
SprayAndWait	0.8832	0.6043	0.8710	2.3585

Table 7: The effect of message lifetime on message delivery success rate.

Algorithm/ttl (min)	30	60	120	240	360	600	900	1200	RECMSR	The normalized RECMSR	Average success rate	Relative promotion ratio (%)
LRDC	0.33	0.34	0.49	0.67	0.78	0.86	0.90	0.94	0.000074	1	0.66	0
DirectDeliver	0.03	0.03	0.14	0.18	0.21	0.32	0.35	0.35	0.00004	0.33	0.20	228
Epidemic	0.35	0.38	0.53	0.71	0.82	0.88	0.94	0.94	0.000023	0	0.7	**-5**
FirstContact	0.12	0.21	0.27	0.32	0.32	0.47	0.5	0.65	0.000036	0.26	0.36	85
SprayAndWait	0.12	0.21	0.35	0.5	0.68	0.77	0.82	0.85	0.00007	0.92	0.54	23
Average relative promotion ratio (%)												82.75

Table 8: The effect of transmission speed on message delivery success rate.

Algorithm/transmitSpeed (kb/s)	12.5	25	50	87.5	125	187.5	250	Average success rate	Relative promotion ratio (%)
LRDC	0.441	0.735	0.882	0.912	0.941	0.941	0.941	0.828	0
DirectDeliver	0.324	0.324	0.353	0.353	0.353	0.353	0.353	0.345	140
Epidemic	0.412	0.735	0.882	0.941	0.941	0.941	0.941	0.828	0
FirstContact	0.412	0.588	0.677	0.647	0.618	0.647	0.647	0.605	36.9
SprayAndWait	0.435	0.653	0.753	0.853	0.853	0.853	0.853	0.75	10.4
Average relative promotion ratio (%)									46.83

The higher the average success rate, the better the routing algorithm performance. For the calculation of relative promotion ratio, we use formula (20) to deal with it. R is any of the four routing algorithms of DirectDeliver, Epidemic, FirstContact, and SprayAndWait. Among them, the average success rate of LRDC is the highest, it can be seen that the performance of LRDC is the best, and the average relative promotion ratio is 46.83%.

6.5. The Effect of BufferSize on LRDC Message Delivery Success Rate. Set other main parameters of simulation environment to $initialEnergy = 50000$, $transmitSpeed = 150$ KB/s, and $msgTtl = 900$ min, the effect of bufferSize on LRDC was tested. The simulation results are as follows. This section presents a set of experiments comparing the performance of LRDC, DirectDeliver, Epidemic, FirstContact, and SprayAndWait and considering the effect of bufferSize on message delivery success rate.

It can be seen from Table 9 that bufferSize has a greater effect on LRDC, Epidemic, and SprayAndWait, while it has less effect on Epidemic and FirstContact. The higher the average success rate, the better the routing algorithm performance. For the calculation of relative promotion ratio, we use formula (20) to deal with it. R is any of the four routing algorithms of DirectDeliver, Epidemic, FirstContact, and SprayAndWait. Among them, the average success rate of LRDC is the highest, it can be seen that the performance of LRDC is the best, and the average relative promotion ratio is 44.43%.

6.6. The Effect of Initial Node Energy on LRDC Message Delivery Success Rate. Set other main parameters of simulation environment to $bufferSize = 100$ MB, $transmitSpeed = 150$ KB/s, and $msgTtl = 900$ min, the effect of *initialEnergy* on LRDC was tested. The simulation results are as follows. This section presents a set of experiments comparing the performance of LRDC, DirectDeliver, Epidemic, FirstContact, and SprayAndWait and considering the effect of *initialEnergy* of nodes on message delivery success rate.

It can be seen from Table 10 that when the *initialEnergy* is less than 20000, all routing algorithms will be greatly affected; when the *initialEnergy* is greater than 20000, the performance of each routing algorithm will gradually stabilize. The lower *initialEnergy* has the greatest effect on the

Table 9: The effect of bufferSize on message delivery success rate.

Algorithm/bufferSize (MB)	5	10	20	40	60	80	100	120	Average success rate	Relative promotion ratio (%)
LRDC	0.405	0.482	0.601	0.748	0.856	0.911	0.941	0.941	0.736	0
DirectDeliver	0.323	0.353	0.353	0.353	0.353	0.353	0.353	0.353	0.349	110.9
Epidemic	0.147	0.235	0.322	0.469	0.606	0.794	0.941	0.941	0.557	32.1
FirstContact	0.303	0.529	0.618	0.618	0.618	0.618	0.618	0.618	0.568	29.6
SprayAndWait	0.415	0.504	0.613	0.723	0.783	0.853	0.853	0.853	0.7	5.1
Average relative promotion ratio (%)										44.43

Table 10: The effect of initial energy of nodes on message delivery success rate.

Algorithm/initial energy	5000	10000	20000	30000	40000	50000	60000	70000	Average success rate	Relative promotion ratio (%)
LRDC	0.453	0.629	0.824	0.882	0.892	0.912	0.941	0.941	0.809	0
DirectDeliver	0.273	0.353	0.353	0.353	0.353	0.353	0.353	0.353	0.343	135.9
Epidemic	0.165	0.271	0 .494	0.682	0.802	0.901	0.941	0.941	0.853	-5.2
FirstContact	0.235	0.411	0.558	0.618	0.618	0.618	0.618	0.618	0.537	50.7
SprayAndWait	0.618	0.725	0.803	0.853	0.853	0.853	0.852	0.853	0.801	1
Average relative promotion ratio (%)										45.6

Epidemic. The average success rate of Epidemic is the highest, and that of LRDC was the second. It can be seen from Table 5 that the average energy consumption of nodes using Epidemic is higher, which is mainly because nodes routing by Epidemic will transmit all messages to all contact nodes, because it will cause a great waste of energy. The higher the average success rate, the better the routing algorithm performance. For the calculation of relative promotion ratio, we use formula (20) to deal with it. R is any of the four routing algorithms of DirectDeliver, Epidemic, FirstContact, and SprayAndWait. Among them, the average success rate of Epidemic is the highest, but it can be seen from Table 5 that the average energy consumption of nodes using Epidemic is higher, and nodes routing by Epidemic will transmit all messages to all contact nodes, so the performance of LRDC is still the best, and the average relative promotion ratio is 45.6.

6.7. Summary of Experiment Results. In the above simulation experiments, we can see that the proposed LRDC algorithm has achieved good experimental results. The reason is that this paper considers the concept of credibility between nodes in the campus opportunistic network, and effectively solves the problem of resource diffusion in the stage of node starvation. The higher credibility between nodes, the more easily the spread resources are accepted and forwarded by each other. It can effectively improve the message delivery rate of the whole network and reduce the redundancy of messages and the delay of message delivery and the energy consumption of nodes.

7. Conclusions

The campus opportunistic networks are that many learner nodes use handheld smart devices and participate in campus collaborative learning. In the campus opportunistic network, the LRDC algorithm is proposed to solve the stability problem of initialization node community and the diffusion problem of learning resources. After a large number of simulation experiments, the comparison results between the proposed LDRC algorithm and the classical algorithms DirectDeliver, Epidemic, FirstContact, and SprayAndWait are as follows: in aspects of average delay and the average hop number, the performance of LDRC is the best; in aspect of RECMSR, LRDC is the highest; in addition, when the transmission speed is the same, LRDC is the best in terms of message delivery success rate, and the average relative promotion ratio is 46.83%. When the buffer size is same, the average success rate of LRDC is also the highest, and the average relative promotion ratio is 44.43%. This paper solves the initialization and stability problem of node community and the diffusion problem of learning resources.

Data Availability

The CRAWDAD dataset Cambridge/haggle (v. 2009-05-29) used to support the findings of this study has been deposited in the https://crawdad.org/cambridge/haggle/20090529 repository ([doi:10.15783/C70011]).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Peng Li, Yuanru Cui, and Qian Liu contributed equally to this work.

Acknowledgments

This work is partly supported by the National Key R&D Program of China under grant no. 2020YFC1523305; the National Natural Science Foundation of China under grant nos. 61877037, 61872228, 61977044, and 62077035; the Key R & D Program of Shaanxi Province under grant nos. 2020GY-221, 2019ZDLSF07-01, and 2020ZDLGY10-05; the Natural Science Basis Research Plan in Shaanxi Province of China under grant nos. 2020JM-302, 2020JM-303, and 2017JM6060; the S&T Plan of Xi'an City of China under grant no. 2019216914GXRC005CG006-GXYD5.1; the Fundamental Research Funds for the Central Universities of China under grant nos. GK201903090 and GK201801004; the Shaanxi Normal University Foundational Education Course Research Center of Ministry of Education of China under grant no. 2019-JCJY009; and the second batch of new engineering research and practice projects of the Ministry of Education of China under grant no. E-RGZN20201045.

References

[1] D. Colomé Cedeño, A. Palmero Ortega, A. Granda Dihigo, and T. Faife Rodríguez, "The collaborative learning of the analysis and modeling of software with the use of Facebook," in *2020 17th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pp. 718–728, Athens, GA, USA, 2020.

[2] C. Liu and K. Wang, "Analysis and modeling of computer-supported collaborative learning system," in *2012 IEEE International Conference on Control Engineering and Communication Technology (ICCECT)*, pp. 1026–1028, Shenyang, Liaoning, China, 2012.

[3] J. Laru, P. Näykki, and S. Järvelä, "Four stages of research on the educational use of ubiquitous computing," *IEEE Transactions on Learning Technologies*, vol. 8, no. 1, pp. 69–82, 2015.

[4] M. Cuka, D. Elmazi, M. Ikeda, K. Matsuo, L. Barolli, and M. Takizawa, "Application of fuzzy logic for IoT node elimination and selection in opportunistic networks: performance evaluation of two fuzzy-based systems," *World Wide Web*, vol. 24, no. 3, pp. 929–940, 2021.

[5] G. Costantino, R. Maiti, F. Martinelli, and P. Santi, "A locality sensitive routing protocol in opportunistic networks with contact profiles," *IEEE Transactions on Mobile Computing*, vol. 19, no. 10, pp. 2392–2408, 2020.

[6] V. Amin and B. David, "Epidemic routing for partially-connected ad hoc networks," *Handbook of Systemic Autoimmune Diseases*, vol. 6, pp. 1–16, 2000.

[7] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[8] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *2004 ACM 5th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 187–198, Roppongi, Tokyo, Japan, 2004.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, 2008.

[10] S. Oramas, V. Ostuni, T. Noia, X. Serra, and E. D. Sciascio, "Sound and music recommendation with knowledge graphs," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 2, pp. 1–21, 2017.

[11] H. Chen, C. Yin, R. Li, W. Rong, Z. Xiong, and B. David, "Enhanced learning resource recommendation based on online learning style model," *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 348–356, 2020.

[12] Y. Hao and Y. Fan, "Mining and prediction of service cooperation relationship of node initialization starvation problem in service system," *Journal of Tsinghua University*, vol. 59, no. 11, pp. 917–924, 2019.

[13] H. Yu and J. Li, "A recommended algorithm to solve the node initialization starvation problem of new projects," *Journal of Software*, vol. 26, no. 6, pp. 1395–1408, 2015.

[14] K. Huang, Y. Fan, and W. Tan, "Recommendation in an evolving service ecosystem based on network prediction," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 906–920, 2014.

[15] Y. Zhang, T. Lei, and Y. Wang, "A service recommendation algorithm based on modeling of implicit demands," in *2016 IEEE International Conference on Web Services (ICWS)*, pp. 17–24, San Francisco, CA, USA, 2016.

[16] G. Wang, J. Zhang, and B. Wang, "Sa-dtn: research of dtn routing based on node social activity," *Computer Application Research*, vol. 28, no. 4, pp. 1524–1526, 2011.

[17] J. Feng, Y. Zhang, and P. Fan, "An adaptive backoff algorithm based on neighbor activity in ad hoc networks," *Journal of System Simulation*, vol. 2008, no. 5, pp. 1348–1352, 2008.

[18] F. Rao, F. Meng, and Y. Xing, "Overlapping community discovery algorithm based on node importance and similarity," *Computer Engineering*, vol. 44, no. 9, pp. 192–198, 2018.

[19] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, pp. 1–20, 2009.

[20] U. Brandes, D. Delling, M. Gaertler et al., "On modularity clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172–188, 2008.

[21] B. H. Good, Y. D. Montjoye, and A. Clauset, "Performance of modularity maximization in practical contexts," *Physical Review E*, vol. 81, no. 4, pp. 1–19, 2009.

[22] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, pp. 1–16, 2004.

[23] S. Li, H. Lou, W. Jiang, and J. Tang, "Detecting community structure via synchronous label propagation," *Neurocomputing*, vol. 151, pp. 1063–1075, 2015.

[24] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset Cambridge/haggle (v. 2009-05-29)," May 2009, https://crawdad.org/cambridge/-haggle/20090529, 10.15783/C70011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3759631, 15 pages
https://doi.org/10.1155/2021/3759631

Research Article

Reinforcement Learning-Based Routing Algorithm in Satellite-Terrestrial Integrated Networks

Yabo Yin,[1] **Chuanghe Huang**,[1] **Dong-Fang Wu**,[1] **Shidong Huang**,[1] **M. Wasim Abbas Ashraf**,[1] **and Qianqian Guo**[2]

[1]*School of Computer Science, Wuhan University, Wuhan 430072, China*
[2]*School of Information Engineering, Zhengzhou Institute of Finance and Economics, Zhengzhou 450053, China*

Correspondence should be addressed to Chuanghe Huang; huangch@whu.edu.cn

Received 9 June 2021; Revised 22 September 2021; Accepted 8 October 2021; Published 28 October 2021

Academic Editor: Yanjie Fu

Satellite-terrestrial integrated network (STIN) is an indispensable component of the Next Generation Internet (NGI) due to its wide coverage, high flexibility, and seamless communication services. It uses the part of satellite network to provide communication services to the users who cannot communicate directly in terrestrial network. However, existing satellite routing algorithms ignore the users' request resources and the states of the satellite network. Therefore, these algorithms cannot effectively manage network resources in routing, leading to the congestion of satellite network in advance. To solve this problem, we model the routing problem in satellite network as a finite-state Markov decision process and formulate it as a combinatorial optimization problem. Then, we put forth a Q-learning-based routing algorithm (QLRA). By maximizing users' utility, our proposed QLRA algorithm is able to select the optimal paths according to the dynamic characteristics of satellite network. Considering that the convergence speed of QLRA is slow due to the routing loop or ping-pong effect in the process of routing, we propose a split-based speed-up convergence strategy and also design a speed-up Q-learning-based routing algorithm, termed SQLRA. In addition, we update the Q value of each node from back to front in the learning process, which further accelerate the convergence speed of SQLRA. Experimental results show that our improved routing algorithm SQLRA greatly enhances the performance of satellite network in terms of throughput, delay, and bit error rate compared with other routing algorithms.

1. Introduction

As the 5th generation communication technologies are widely used, terrestrial network can provide high bandwidth and low delay communication services to the users within the coverage of base stations [1]. However, for those remote areas where base stations are not deployed or where base stations are destroyed by natural disasters, terrestrial network usually cannot meet the communication needs of users. Infrastructure of satellite network is rarely damaged by natural disasters, and it has wide coverage [2–4], so it is usually regarded as an essential component of terrestrial network. Satellite-terrestrial integrated network (STIN) has wide coverage and high flexibility and is able to compatible with the existing 5G network. Thus, it is a reliable paradigm to provide the Internet services and is receiving much attention from researchers [5–8]. In particular, when users are unable to communicate through terrestrial network in aviation or navigation, the STINs can provide them with communication services.

Satellite routing algorithm is an important technique in STINs, and there are much researches on it. Considering that the number of satellites is small and the structure of traditional satellite network is simple, existing satellite routing algorithms are developed from the routing algorithms of terrestrial network, such as OSPF [9], RIP [10], and AODV [11]. However, most algorithms are depended on the shortest path or minimum cost. Therefore, satellite network is prone to

congestion in the process of routing. In addition, with the expansion of satellite network (e.g., Starlink), these routing algorithms cannot converge quickly in the limited communication time, which seriously degrades the communication performance of satellite network and wastes the communication resources of satellites at the same time. On the other hand, most of the work on satellite routing ignores the impact of user request resources on the performance of satellite network. Liu et al. [12] proposed a fragment-based load balancing route scheme to control the traffic of LEO satellite network. Qi et al. [13] improved the quality of service (QoS) of users by jointly optimizing the rate and routing in LEO satellite network. Considering the traffic distribution density in different areas, the authors in [14] proposed a distributed routing algorithm based on traffic prediction. However, the above algorithms do not take into account the impact of the current user's routing on the subsequent user's routing, resulting in the network performance is degraded. In addition, considering the power and computing resources of LEO satellite, it is not appropriate to deploy these algorithms on satellites. Therefore, it is very challenging to design an efficient satellite routing algorithm.

Recently, machine learning and deep learning [15, 16] have been used extensively. Some researchers began to use these methods to solve network communication problems [17–19]. The authors in [20] regarded satellite network topology as a series of snapshots and used particle swarm optimization algorithm for routing in each snapshot. However, deep learning is an approximate algorithm, which is not suitable for sequential decision problems. Reinforcement learning is a method based on trial and error. In the process of learning, the agent interacts with the environment and gets a corresponding reward. And this reward guides the agent to find the best strategy. Moreover, reinforcement learning is very suitable for dealing with sequential decision problems and achieves better results than human beings [21], and it has been applied in resource allocation [22], capacity management [23], and combinatorial optimization [24, 25]. Compared with other reinforcement learning algorithms, Q-learning is a simple and efficient reinforcement learning method, which has a fast convergence speed. In addition, it is very suitable for solving discrete problems. Inspired by the above references, we regard the satellite routing problem as a turn-base game and model it as a finite-state Markov decision process. Then, we put forward a Q-learning-based routing algorithm to solve satellite routing problems.

In this paper, we mainly investigate the satellite routing problem in STINs. Choosing a path from source node to destination node can be regarded as a turn-based game. And this is a finite-state Markov decision process. So we model the routing problem as a Markov decision process and define its state space, action space, and reward function. We propose QLRA algorithm to make full use of satellite network resources and improve the quality of service of users. In addition, in order to accelerate the convergence speed of QLRA algorithm, we propose a split-based speed-up convergence strategy and design a speed-up Q-learning-based routing algorithm (SQLRA). Moreover, we update the Q value from back to front to further improve the speed of SQLRA algorithm. The contributions of this paper are summarized as follows:

(1) We model the satellite routing as a Markov decision process and define its action and state spaces and reward function. And we propose a Q-learning-based satellite routing algorithm (QLRA). QLRA algorithm can select the optimal paths according to the current states of satellite network and the users' request resources when routing

(2) Aiming at the slow convergence speed of QLRA algorithm, we analyse the problem and propose a split based speed-up convergence strategy to accelerate the convergence speed of QLRA. Based on QLRA algorithm, we design a speed-up Q-learning-based routing algorithm (SQLRA). In addition, we update the Q-value from back to front to further improve the convergence speed of SQLRA algorithm. Experimental results show that SQLRA algorithm converges faster than QLRA algorithm

(3) Our proposed algorithm SQLRA can make full use of network resources while meeting the requirements of users. Numerical simulation results show that SQLRA algorithm effectively enhances the network performance compared with other algorithms

The remainder of this paper is organized as follows. In Section 2, the related work is reviewed. In Section 3, we introduce network model and problem formulation. In Section 4, the satellite routing algorithm based on Q-learning and the speed-up Q-learning-based routing algorithm are presented. In Section 5, we evaluate the performance of SQLRA algorithm in two different scenarios, analyse, and discuss the experimental results. Section 6 concludes this paper and gives future research issues.

2. Related Work

There are much researches on satellite network routing issues. In order to reduce the link congestion and the imbalance of load distribution, Liu et al. [26] proposed an iterative Dijkstra algorithm to optimize satellite communication path. The traditional LEO satellite network ignores the delay of links in routing, which leads to the incomplete evaluation of satellite network performance. To solve this problem, in [27], a satellite routing algorithm taking delay into account was proposed. The authors in [28] proposed a routing algorithm based on cooperative game theory to solve the problem of propagation delay and traffic load imbalance in LEO satellite network. Jiang et al. [29] designed a routing algorithm based on fuzzy theory to meet the multilevel needs of users. By leveraging orbit prediction information, Pan et al. [30] put forward a dynamic on-demand routing scheme to reduce the routing convergence and the communication overhead. Hao et al. [31] proposed a routing strategy based on energy-aware and load-balancing to meet the different communication services of users.

As a reliable communication paradigm, much work has been done on STINs. In order to improve the power utilization of satellites, the authors in [32] proposed a data offloading scheme in STINs to jointly allocate the power and resources of satellites. Zhang et al. [33] used edge computing techniques to improve the QoS of STINs. In order to reduce the cost of gateway deployment and data routing in STINs, the authors in [34] proposed a joint satellite gateway deployment and routing scheme. Xu et al. [35] proposed a hybrid routing algorithm to realize the seamless integration of STINs. The authors in [36] presented an end-to-end routing method based on heuristic strategy to improve the QoS of STINs.

Reinforcement learning is an effective method to cope with sequential decision problems, and it has been applied in many fields. Liu et al. [37] used Q-learning to implement the content caching problem in dynamic cloud content distribution network. To improve the efficiency of the Internet of Things, Pan et al. [38] used Q-learning to identify blocked links. A Q-learning method is proposed in [39] to improve the network performance and reduce the energy consumption of wireless sensor networks. Qiao et al. [40] proposed a joint optimization scheme of cache content placement and bandwidth resource allocation based on deep reinforcement learning in the Internet of Vehicles. Q-learning technique is widely used, and there are few researches on routing using Q-learning technique in satellite network. In this research, we use reinforcement learning to solve the routing problem in satellite network.

3. System Model and Problem Formulation

3.1. Network Model. The STINs used in this paper are shown in Figure 1. The STINs are composed of a terrestrial network and a LEO satellite network. The terrestrial network consists of base stations, routers, satellite gateways, and user terminals, and the satellite network consists of a large number of LEO satellites. The terrestrial network is able to connect with the satellite network with the aid of satellite gateways. Considering the high-speed mobility of the satellites in satellite constellation, each satellite is only connected to its neighbour satellites or satellites in its adjacent orbits. The communication link between satellites is bidirectional, and the specific structure is shown in Figure 2.

When users communicate with their peers, the system first determines whether to reach their peers through the terrestrial network. Specifically, if their peers can be reached through the terrestrial network, then the data will be transmitted directly to their peers through the terrestrial network. Otherwise, the terrestrial network will transmit the data to the LEO satellites through the satellite gateways and then retransmit the data to their peers through the LEO satellite network. With the increasing number of satellites, existing satellite routing algorithms become unsuitable, which seriously degrade the performance of satellite network. Therefore, we focus on the satellite routing problem in this paper.

Considering that the topology of satellite network changes with time, inspired by reference [20], here we divide the whole operation time T of satellite network into N_T time slices, and the duration time of each time slice is T_t. We assume that the satellite network topology is fixed in each time slice. So the total time can be obtained by

$$\sum_{t=1}^{N_T} T_t = T. \tag{1}$$

The number of snapshots is related to the number of orbits and the number of satellites in each orbit. The time interval of the snapshot T_t is related to the inclination of the orbits. The smaller the time interval, the higher the accuracy of the snapshot. If the time interval is small, a large number of topologies will be generated, which leads to the complexity of the network structure. In practice, the time interval is no more than the minimum visible time of the satellite links. We define T_t as

$$T_t \le \min\{\tau(u, v)\}, u \ne v, u, v \in V, \tag{2}$$

where $\tau(u, v)$ represents the visible time between satellite u and satellite v. Here, we set the time interval T_t to 4 minutes.

Here, we use undirected graph $G = (V, E)$ to represent satellite network topology, where V represents the set of satellites, $V = \{1, 2, \cdots, N\}$, and N is the number of satellites. And E is the set of links between satellites. Here, we assume that the network structure is a connected graph. We define it as

$$E = \{\text{link}(u, v)\}, u \ne v, u, v \in V, \tag{3}$$

where $\text{link}(u, v)$ represents the link between satellite u and satellite v. And satellite v is a neighbour of satellite u. Considering that the state of satellite link consists of many parameters, we redefine link $\text{link}(u, v)$ as

$$\text{link}(u, v) = (\text{bandwidth}, \text{delay}, \text{error}, \text{time}), \tag{4}$$

where variables bandwidth, delay, error, and time represent the available bandwidth, the propagation delay, the bit error rate, and the available time of link $\text{link}(u, v)$, respectively. In addition, considering the duration of each time interval is short, we assume that the communication time of each user is greater than the duration of each time slice. We define it as

$$u^{\text{req\_}t} \ge T_t. \tag{5}$$

When transmitting data, it is necessary to find an optimal path from source satellite to destination satellite according to the current link states of satellite network. We assume that satellite 0 is source satellite and satellite 5 is destination satellite in Figure 3. There are multiple paths from satellite 0 to satellite 5. However, with a large number of users accessing to satellite network, satellite network resources are exhausted due to the load imbalance, which leads to the congestion of satellite links in advance. Therefore, the performance of satellite network is seriously degraded. For example, in the beginning, the optimal path

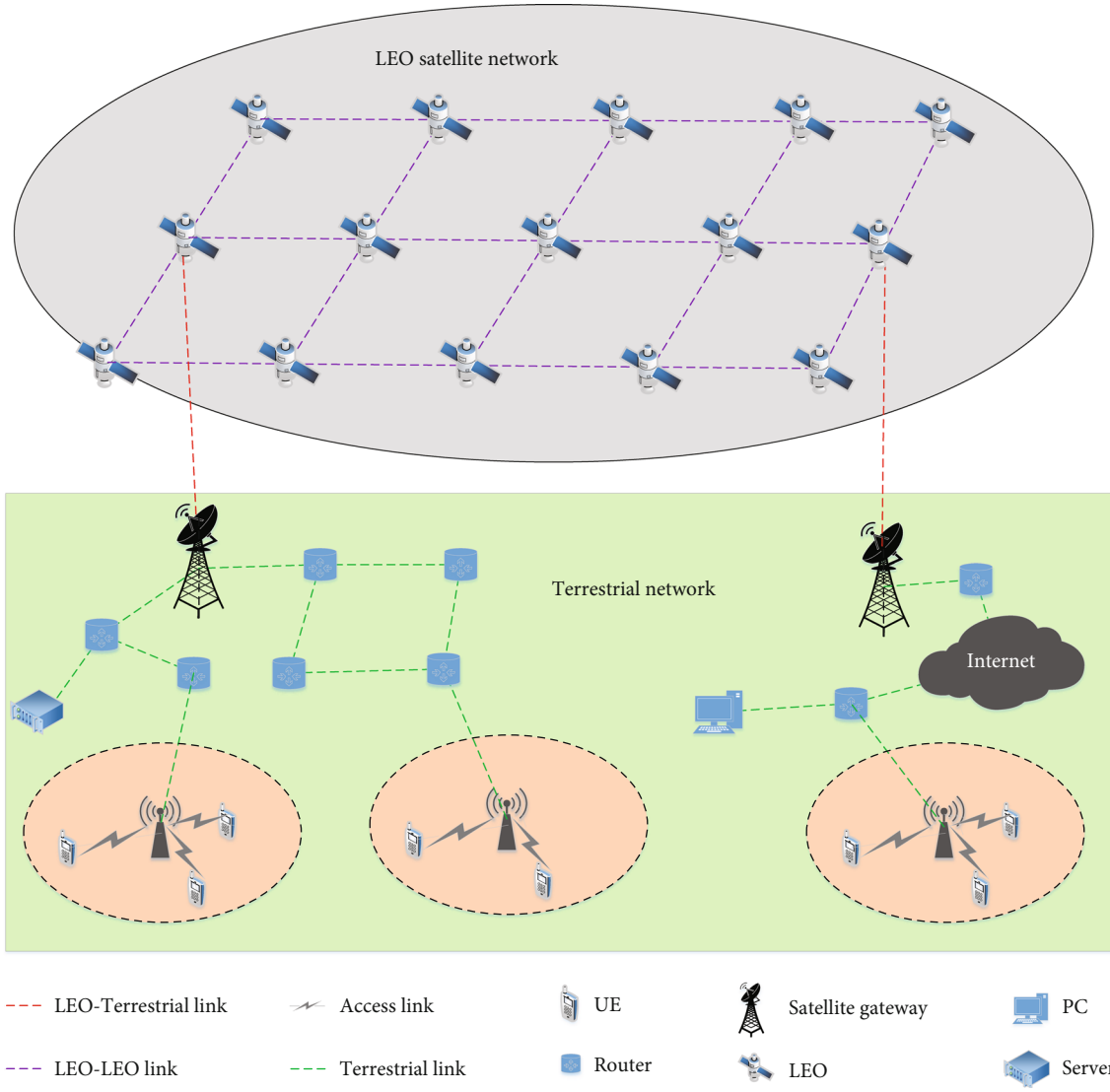


Figure 1: Structure of satellite-terrestrial integrated networks.

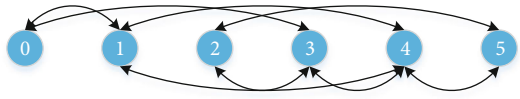


Figure 2: Communication links of LEO satellite network.

from satellite 0 to satellite 5 is 0-3-4-5 in Figure 3. With the increase of users' number, link 0-3 is congested because of the consumption of bandwidth resources, resulting in the next user cannot choose link 0-3 in the optimal path. Therefore, the path from satellite 0 to satellite 5 changes from path 0-3-4-5 to path 0-1-4-5 and finally to path 0-1-4-3-5. The specific process of path changing is shown in Figure 3.

In Figure 3, the dotted lines with different colours represent different selected paths. From Figure 3, we see that the optimal path from satellite 0 to satellite 5 changes gradually with the consumption of communication resources of satellite links.

3.2. Problem Formulation. We assume that the bandwidth capacity of link $\text{link}(u, v)$ is $c(u, v)$, variable u_i^{req} represents the bandwidth resource requested by the ith user. Before transmitting data, we need to find a path from source satellite v_s to destination satellite v_d for the ith user. The path is defined as

$$\text{path}(v_s, v_d) = \{\text{link}(v_s, u), \cdots, \text{link}(v, v_d)\}, u, v \in V, u \neq v. \tag{6}$$

Here, y is an indicator function which indicates whether there is a link in the selected path. If satellite link $\text{link}(u, v)$ is in the selected path, then $y(\text{link}(u, v)) = 1$; otherwise, $y(\text{link}(u, v)) = 0$.

Here, we use functions $B(x)$, $D(x)$, $E(x)$, and $T(x)$ to represent the average bandwidth, the delay, the bit error rate, and the available time of the user in path x, respectively.

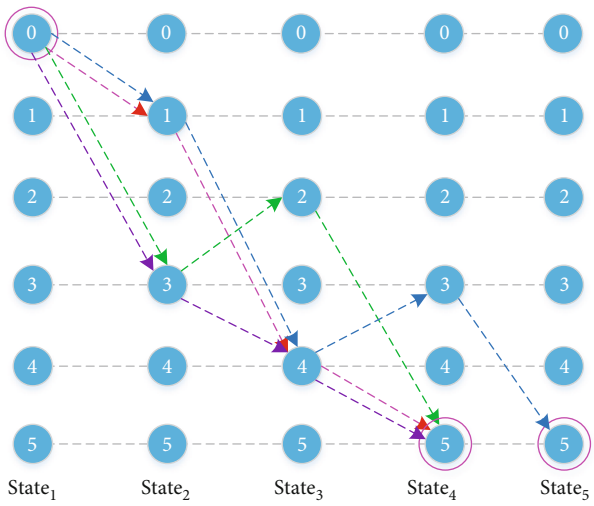


Figure 3: Process of path changing in satellite network.

Variable path^i represents the path of the ith user from source satellite to destination satellite.

$$B\left(\text{path}^i\right) = \frac{\sum \text{band}(\text{link}(u,v))}{\text{length}\left(\text{path}^i\right)}, \text{link}(u,v) \in \text{path}^i, \quad (7)$$

$$D\left(\text{path}^i\right) = \sum \text{delay}(\text{link}(u,v)), \text{link}(u,v) \in \text{path}^i, \quad (8)$$

$$E\left(\text{path}^i\right) = \sum \text{error}(\text{link}(u,v)), \text{link}(u,v) \in \text{path}^i, \quad (9)$$

$$T\left(\text{path}^i\right) = \min\left(\text{time}(\text{link}(u,v))\right), \text{link}(u,v) \in \text{path}^i, \quad (10)$$

where function $\text{length}(\text{path}^i)$ is the length of the path path^i and function $\text{band}(\text{link}(u,v))$ is the bandwidth of link $\text{link}(u,v)$. Similarly, functions $\text{delay}(\text{link}(u,v))$, $\text{error}(\text{link}(u,v))$, and $\text{time}(\text{link}(u,v))$ are the delay, the bit error rate, and the available time of link $\text{link}(u,v)$, respectively.

Our goal is to maximize the utility of all users by considering the bandwidth, the delay, the bit error rate, and the available time of network links in the process of routing.

$$\frac{1}{M} \max\left(\sum_{i=1}^{M} \theta \cdot B\left(\text{path}^i\right) + \beta \cdot D\left(\text{path}^i\right) + \lambda \cdot E\left(\text{path}^i\right) + \omega . T\left(\text{path}^i\right)\right), \quad (11)$$

$$\text{subject to } 0 \le \sum_{i=1}^{M} u_i^{\text{req}} \cdot y(\text{link}(u,v)) \le c(u,v), v \in V, u \ne v, \quad (12)$$

$$y(\text{link}(u,v)) \in \{0,1\}, \quad (13)$$

$$\sum_{u \in V, u \ne v} y(\text{link}(u,v)) \cdot u_i^{\text{req}} = \sum_{u \in V, u \ne v} y(\text{link}(v,u)) \cdot u_i^{\text{req}}, \forall \text{v} \in \text{V}, \text{v} \notin \{\text{v}_s, \text{v}_d\}, \quad (14)$$

$$\theta + \beta + \lambda + \omega = 1, \quad (15)$$

where M is the number of users accessing to satellite network. Equation (12) ensures that the bandwidth resource requested by users is less than the total bandwidth resources of each link. Equation (13) is an indicator function which indicates whether link $\text{link}(u,v)$ is in the selected path. If link $\text{link}(u,v)$ is in the selected path, $y(\text{link}(u,v) = 1$; otherwise, $y(\text{link}(u,v)) = 0$. Equation (14) is used to ensure that for any intermediate link, the incoming traffic and the outgoing traffic are equal. Equation (11) is a combinatorial optimization problem, and we use reinforcement learning to solve it. In the experiment, we use analytic hierarchy process (AHP) to judge the influence of the weight of each parameter on the performance of the satellite network [41].

4. A Satellite Routing Algorithm Based on Reinforcement Learning

Reinforcement learning is mainly composed of the agent and the environment. The agent interacts with the environment and learns the optimal strategy according to the feedback of environment. In particular, the reinforcement learning framework is shown in Figure 4. In the current state s_t, the agent chooses an action a_t according to the policy π and execute it. And the environment returns a corresponding reward to the agent, and the environment moves its state from s_t to the next state s_{t+1}. The agent interacts with the environment continuously until the episode is end or the number of interaction steps reaches the threshold set in advance.

In STINs, the environment is the link states of satellite network, and it is time-varying. And the agent is deployed in ground control center. In route discovery phase, the agent chooses a valid action according to the users' request and jumps to the satellite whose index is the valid action value. The environment gives the agent a corresponding reward, and the link states of satellites are changed simultaneously. The agent interacts with the environment until a path from source satellite to destination satellite is selected. The routing process is modelled as Markov decision processes (MDPs) and represented by $M = (S, A, P, R)$, where S is state space, A is action space, and R is reward value. Furthermore, P is the state transition probability function, $P(s' \mid s, a) = P(s' = s' \mid s = s, a = a)$. The specific details are defined as follows:

(1) *State Space*. The satellite link state considered in this paper includes available bandwidth, propagation delay, bit error rate, and available time. We define the state of link as

$$\text{link}_{i,j} = \left(b_{i,j}, d_{i,j}, e_{i,j}, t_{i,j}\right), 0 \le i, j \le N, \quad (16)$$

where N is the number of satellites. And $\text{link}_{i,j}$ denotes the

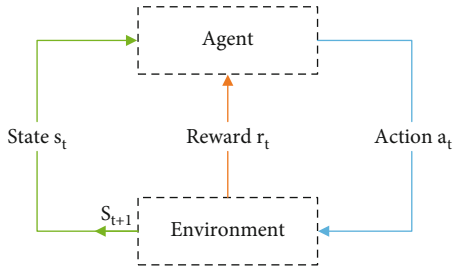


Figure 4: Reinforcement learning framework.

link state between satellite i and satellite j. In addition, variables $b_{i,j}$, $d_{i,j}$, $e_{i,j}$, and $t_{i,j}$ represent the available bandwidth, the propagation delay, the bit error rate, and the available time of link between satellite i and satellite j, respectively. The variable S_t represents the states of all links in satellite network.

$$S_t = \left\{\text{link}_{i,j}\right\}, 0 \le i, j \le N, j \in N(i), \tag{17}$$

where $N(i)$ represents the set of neighbours of satellite i. And variable S_t is the environment of reinforcement learning.

(2) *Action Space*. In satellite network, the action is used to describe the process of the agent moving from one satellite to another. For example, taking action a, the agent moves from the current satellite to the satellite whose index is a. The number of satellites is N, so the action set is denoted by $A = \{1, 2, \cdots, N\}$. For the convenience of calculation, the action is coded by one-hot coding in our simulations

(3) *Reward Value*. The rewards are used to motivate the agent to search for the optimal strategy. The agent obtains the rewards by the states of satellite links. Different link states give different rewards. In order to avoid the impact of different rewards on the accuracy of results, we use min-max operation to normalize them. Function max (b) is the maximum of variable b, and function min (b) is the minimum of variable b. We elaborate the specific operation below:

$$r(b) = \frac{b - \min(b)}{\max(b) - \min(b)}, \tag{18}$$

$$r(d) = \frac{\max(d) - d}{\max(d) - \min(d)}, \tag{19}$$

$$r(e) = \frac{\max(e) - e}{\max(e) - \min(e)}, \tag{20}$$

$$r(t) = \frac{t - \min(t)}{\max(t) - \min(t)}, \tag{21}$$

where $r(b)$ is the reward generated by the bandwidth of the selected link. Similarly, $r(d)$, $r(e)$, and $r(t)$ represent the rewards generated by the delay, the bit error rate, and the available time of the selected link, respectively. The link delay and the bit error rate are negative to the link selection. Therefore, we use monotone decreasing function to present the corresponding rewards in the process of normalization. In this way, the total reward generated by the selected links is shown in Equation (22).

$$r = \theta \cdot r(b) + \beta \cdot r(d) + \lambda \cdot r(e) + \omega \cdot r(t), \tag{22}$$

where variables θ, β, λ, and ω are weight coefficients, respectively, which are used to represent the importance of each reward. Here, we use analytic hierarchy process (AHP) to determine the value of these parameters. In addition, variable R_t is the cumulative reward that the agent gets by taking action a_t in current state s_t. We define it as

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}. \tag{23}$$

We use state-action value $Q(s, a)$ to represent the cumulative reward value obtained by the agent taking action a_t in current state s_t. And this value indicates the quality of each action in the current state. We define it as

$$Q_\pi(s_t, a_t) = E_\pi[R_t \mid S_t = s_t, A_t = a_t]. \tag{24}$$

Choosing different strategy functions will get different state-action values. Our goal is to find the best strategy function to make the agent choose the appropriate action in each state.

$$Q_\pi{}^*(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t). \tag{25}$$

According to Equation (25), we maximize the state-action value $Q(s, a)$ to find the optimal strategy. The strategy with the symbol "$*$" is the optimal strategy. At last, the best action in each state can be selected by searching Q-table.

4.1. Path Checking Algorithm. When the communication resources of satellite links are exhausted, the links will be congested. If there is not a path from source node to destination node in satellite network, the routing algorithm cannot find a suitable path to destination node. In order to avoid this situation, we first judge whether there is a reachable path to destination node before looking for a path. If there is no such a path, it means that links are congested or disconnected in satellite network. At this time, the routing algorithm stops looking for paths, reducing the waste of computing resources. We use pseudo code to describe the details of path checking algorithm.

where $flag = 1$ indicates that there is a path from *start_node* to *end_node* and $flag = 0$ indicates that there is no path from *start_node* to *end_node*.

4.2. Q-Learning-Based Routing Algorithm (QLRA). Q-learning is a reinforcement learning method based on value function, and it uses a Q-table to store different state-action

```
Input: start_node, end_node, graph.
Output: the flag which indicates whether there is a valid path from start_node to end_node.
1.Initialize flag = 0, path ={}.
2.Get the neighbours of start_node based on the network structure graph, neighbours_list.
3.Let path = path∪{start_node}.
4.while neighbours_list:
5.   Pop a node from neighbours _list, node.
6.   if node not in path:
7.      if node == end_node:
8.         path = path ∪{node}.
9.         flag =1.
10.     end if
11.     else:
12.           Get the neighbours of node, node_neighbours.
13.           Add the node_neighbours to the list neighbours_list .
14.           path = path ∪{node}.
15.     end else
16.  end if
17.end while
```

Algorithm 1: Path checking algorithm (PCA).

values. Because Q-learning is a mode-free method, it does not need prior knowledge in the process of learning. In addition, Q-learning learns the optimal strategy by trial and error, and it is very suitable for the dynamic satellite network. Therefore, here we try to use Q-learning to select the optimal route. The main idea of Q-learning algorithm is we first initialize the Q-table, then select an action from the current state by ε-greedy strategy, and the environment moves from the current state to the next state and update the state-action value $Q(s, a)$ by using Bellman Equation. At last, this process is repeated until the Q-table converges. The Bellman Equation is defined as

$$Q(s,a) = Q(s,a) + \alpha\left[r + \lambda \max Q\left(s', a'\right) - Q(s,a)\right], \quad (26)$$

where α denotes the learning rate, λ is the discount rate, and s' represents the next state. Furthermore, variable r denotes the immediate reward obtained from the environment.

$$a = \begin{cases} \text{select an action randomly}, & r' < \varepsilon, \\ \arg\ \max_a Q(s,a), & r' \geq \varepsilon. \end{cases} \quad (27)$$

In the process of training, we use ε-greedy strategy, as shown in Equation (27), to avoid the result falling into the local optimal solution. The strategy can achieve the trade-off between exploration and exploitation, where r' represents the random number generated in the process of selecting the action and ε represents the probability of action exploration. Furthermore, in order to speed up the convergence of Q-table, we redefine the reward function, where C is a constant.

$$r(s_{t+1} \mid s_t, a_t) = \begin{cases} r(s_t, a_t) & s_{t+1} \text{ is not terminal state}, \\ r(s_t, a_t) + C & s_{t+1} \text{ is the terminal state}. \end{cases} \quad (28)$$

We first judge whether there is a feasible path. If there is a path, the optimal path is selected by means of Q-learning routing algorithm. Then, the reward matrix and structure of satellite network are updated. The specific Q-learning-based routing algorithm is described as follows:

where the function *PCA* in QLRA represents the path checking algorithm proposed above.

4.3. Speed-Up Q-Learning-Based Routing Algorithm (SQLRA). In the process of selecting the next hop, the agent will jump from the current state to the previous state. And this operation will result in some repeated and invalid sequences in the selected path. When selecting the path from source satellite node B to destination satellite node J, there will be some repeated and invalid sequences. The specific details are shown in Figure 5. For example, in path $B->E->F->C->B->E->H->G->H->G->H->G->J$, the sequences in the magenta dashed box indicate that a routing loop has occurred, and the sequences in the blue dashed box indicate that a ping-pong effect has occurred. From Figure 5, we can see that sequences $E->F->C->B$ and $G->H$ are repeated and invalid. These repeated and invalid sequences will not only waste computing resources but also lead to the slow convergence speed of QLRA algorithm.

Although QLRA algorithm uses ε-greedy strategy to select effective actions in the learning process, it still generates invalid sequences. To avoid the routing loop or ping-pong effect, we must prevent the agent from jumping from the current state to the previous visited state, when the agent selects an effective action. To solve this problem, we propose a split-based speed-up convergence strategy. The specific split process is shown in Figure 6.

Similar to the broadcast mechanism, we split the satellite network according to the neighbour information of nodes. As shown in Figure 6, for destination node J, we regard its neighbour nodes as the first layer, and nodes with the same colour belong to the same layer. We update the Q value of all

Input: *start_node*, *end_node*, *graph*, *R*, *users_req_list*.
Output: the optimal paths.
1. Initialize Q-table, α, γ, ε, Episodes=*M*.
2. for user_req in users_req_list:
3. *flag* = *PCA* (*start_node*, *end_node*, *graph*).
4. if *flag* ==1:
5. for *i*=1 to Episodes:
6. *current_state* = *start_node*.
7. while *current_state*!= *end_node*:
8. Select the action *a* based on Eq. (27).
9. Get the corresponding reward value generated by each parameter according to Eq. (18), (19), (20) and (21).
10. Obtain the total reward *r* based on Eq. (22).
11. The agent move to the next state *s'*.
12. Update the Q-table based on Eq. (26).
13. Let *current_state* = *s'*.
14. end while
15. end for
16. Select the optimal path *path* from the converged Q-table based on Eq. (25).
17. Update the reward matrix *R* based on the consumption of link resources.
18. Update the structure of satellite network *graph* based on the consumption of link resources.
19. end if
20. else:
21. There is no path from *start_node* to *end_node*.
22 Break
23. end else
24.end for

Algorithm 2: Q-learning-based routing algorithm (QLRA).

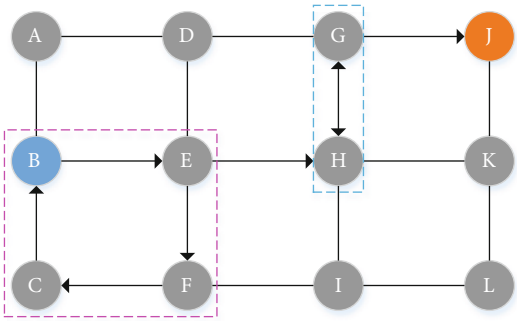


Figure 5: Repeated and invalid sequences during routing.

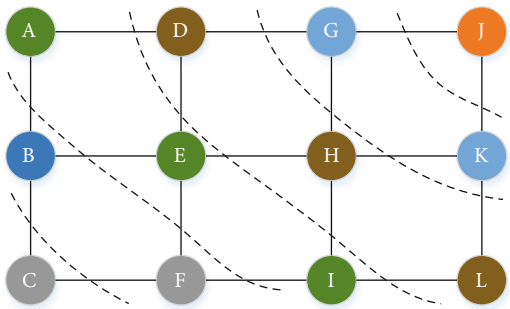


Figure 6: Split process of satellite network.

nodes in the same layer every time until all nodes of satellite network are updated. This scheme ensures that the *Q* value of each node is updated along a horizontal direction, and it destroys the condition of forming a loop between nodes. Therefore, the split operation accelerates the convergence speed of QLRA algorithm.

The traditional reinforcement learning updates the *Q* value of each node from front to back. But in the satellite network, we can obtain all the states of the satellite network in advance. Therefore, no matter what state the agent is in, we can know the next state of agent according to the actions taken by the agent. In addition, according to Equation (26), we know that updating the node's *Q* value from back to front make Q-table converge faster. Figure 7 illustrates how the agent updates the *Q* value of each node. Based on QLRA algorithm, we propose SQLRA algorithm with speed-up convergence strategy. The specific pseudo code of SQLRA is as follows:

where the function *BFS* in SQLRA represents the breadth first search algorithm. We search from the last node *end_node* to get the traversal sequences of the whole network. And we use function *Neighbour* to get the neighbour information of each node from the satellite network structure.

Figure 8 shows the convergence speed of SQLRA and QLRA. We observe Figure 8 that SQLRA converges faster than QLRA. We also observe that SQLRA needs 30 episodes to converge, and QLRA needs 60 episodes to converge. The main reason is that in the process of routing, our split-based speed-up convergence strategy reduces the invalid sequences. In addition, we update the *Q* value of the nodes from back to front, which further accelerate the convergence speed of SQLRA.

Input: *start_node*, *end_node*, *graph*, *R*, *users_req_list*.
Output: the optimal paths.
1. Initialize Q-table, α, γ, Episodes=*M*.
2. for user_req in users_req_list:
3. *flag* = PCA (*start_node*, *end_node*, *graph*).
4. if *flag* ==1:
5. *nodes_list* = ***BFS***(*graph*, *end_node*)
6. for *i*=1 to Episodes:
7. for *node* in *nodes_list*:
8. if *node* != *end_node*:
9. *neighbours* = ***Neighbour***(node).
10. for *neighbour* in *neighbours*:
11. Get the corresponding reward value generated by each parameter according to Eq. (18), (19), (20) and (21).
12. Obtain the total reward *r* based on Eq. (22).
13. Update Q(node, neighbour) based on Eq.(26).
14. end for
15. end if
16. end for
17. end for
18. Select the optimal path *path* from the converged Q-table based on Eq. (25).
19. Update the reward matrix *R* based on the consumption of link resources.
20. Update the structure of satellite network *graph* based on the consumption of link resources.
21. end if
22. else:
23. There is no path from *start_node* to *end_node*.
24 Break
25. end else
26.end for

Algorithm 3: Speed-up Q-learning-based routing algorithm (SQLRA).

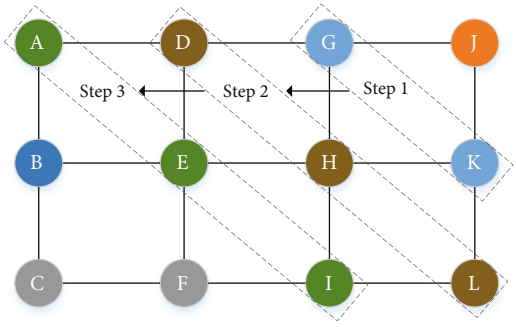


Figure 7: Mechanism for updating *Q* value from back to front.

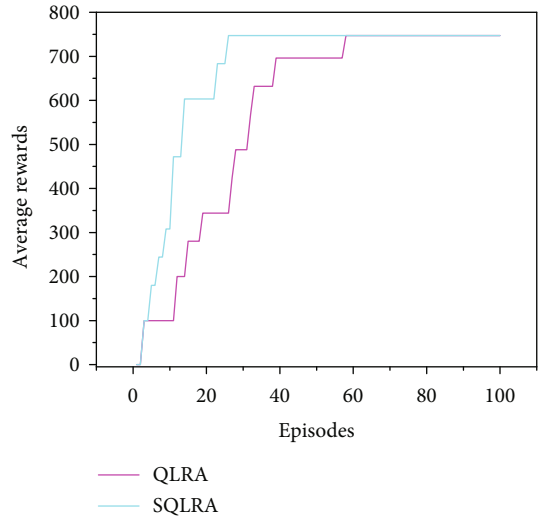


Figure 8: Convergence speed of QLRA and SQLRA with different episodes.

5. Performance Evaluation

In this section, we verify the effectiveness of the proposed algorithm SQLRA. First, the simulation environment and related parameters are introduced. Then, we compare SQLRA with QLAODV [42], QSR [43], OSPF [30], and ACO [44] in the performance of throughput, delay, bit error, and visible time. At last, we analyse and discuss the simulation results.

5.1. Experimental Parameter Settings. We conduct numerical simulations to verify the effectiveness of our proposed routing algorithm SQLRA. As for the satellite network used in this paper, we use satellite tool kit (STK) to simulate it. The satellite constellation adopts the Walker delta model. The satellite network consists of eight orbital planes, and every orbit has six satellites. There are 48 LEO satellites in total. The inclination angle of each satellite orbit is 45 degrees, and the altitude of satellite orbit is 650 km. Each satellite is only connected to its particular neighbour satellites. Please refer to Section 3 for more details. Due to the long

Table 1: Parameters of satellite constellation.

Parameters	Value
Number of satellites	48
Number of orbits	8
Orbit inclination	45°
Satellite number per orbit	6
Altitude of orbit	650 km
Simulation time	1400 s
Right ascension of ascending node	25°
Neighbours of a satellite	4

distance between satellites, the delay of satellite communication is mainly determined by the propagation delay of satellite links. Therefore, we mainly consider the propagation delay of satellite links in this paper. Furthermore, the propagation delay of satellite links in the same orbit is also different. For simplicity, we assume that the bit error rate of each satellite link follows a uniform distribution. And the bandwidth resources requested by each user follows a Poisson distribution. The specific parameters of satellite constellation are shown in Table 1. In our simulations, we use Pycharm as development tool. The environment is Win10 Operating System with 16 G RAM and 3.2 GHz CPU.

For QLRA and SQLRA, the learning rate α affects the convergence speed of algorithms. And the discount factor represents the impact of future rewards on the current result, which can prevent the agent from falling into the local optimum. The discount factor is between 0 and 1. The higher the value is, the more critical the future reward is. Through the analysis of experimental results, when the learning rate α and discount factor γ are set to 0.001 and 0.9, respectively, the convergence effect of the algorithms is the best. At this time, the convergence results of QLRA and SQLRA are the same. The weight of each parameter in the reward value can be obtained by analytic hierarchy process. Here, we set the values of θ, β, λ, and ω in Equation (11) as 0.30, 0.15, 0.18, and 0.37, respectively.

5.2. Results Analysis and Discussion. In this section, we evaluate the performance of SQLRA algorithm in two different scenarios: one is that all users communicate with each other by the same source satellite node and destination satellite node, and the other is that all users communicate with each other through different source satellite nodes and destination satellite nodes.

(1) Performance in communication scenario with same node pair

The users' requests in this scenario have the same source satellite node and destination satellite node. Because the source satellite node and destination satellite node of the selected paths are fixed, here we use average throughput, average delay, average bit error rate, and average visible time to measure the performance of routing algorithms. We compare the performance of algorithms with varying numbers of users.

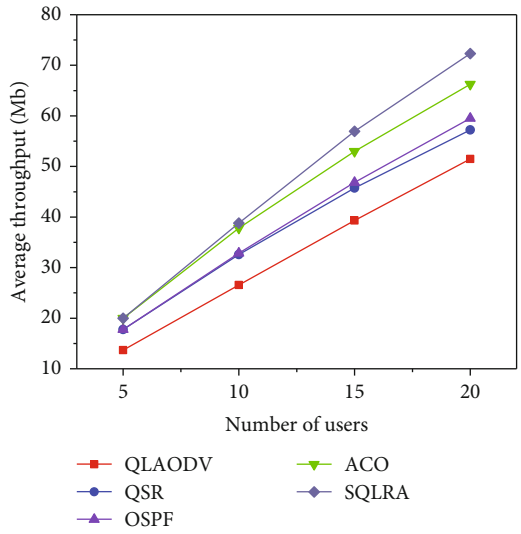


Figure 9: Average throughput with varying number of users.

(A) Average throughput analysis

Figure 9 shows the average throughput of different algorithms with varying number of users. From Figure 9, we can see that as the number of users increases, the average throughput obtained by all algorithms is increasing. At the same time, we can draw the following conclusions from Figure 9. First, QLAODV algorithm has the worst performance. Compared with AODV, QLAODV considers not only the number of hops and the delay but also the bit error rate and the available bandwidth. However, QLAODV still prefers to select the path with fewer hops when selecting the next hop. In satellite network, the distance between satellites in the same orbit is different from that between satellites in different orbits; the path with the minimal hops is not always the optimal path. Second, although both ACO and OSPF consider the same characteristics of satellite links in the process of routing, the performance of ACO is better than that of OSPF. The main reason is that OSPF is based on greedy strategy and is easy to fall into the local optimum, while ACO algorithm tries to find the global optimum as much as possible by using the positive feedback mechanism. Lastly, compared with other routing algorithms, our proposed SQLRA has the best performance. The reason is that in the process of path selection, SQLRA not only considers the states of current satellite links but also considers the impact of future rewards on the current selected links. In addition, the Q-table of SQLRA can converge after a certain number of iterations. Therefore, SQLRA mostly finds the optimal solution.

(B) Average delay analysis

We show in Figure 10 the average delay with different number of users. We observe that with the number of users

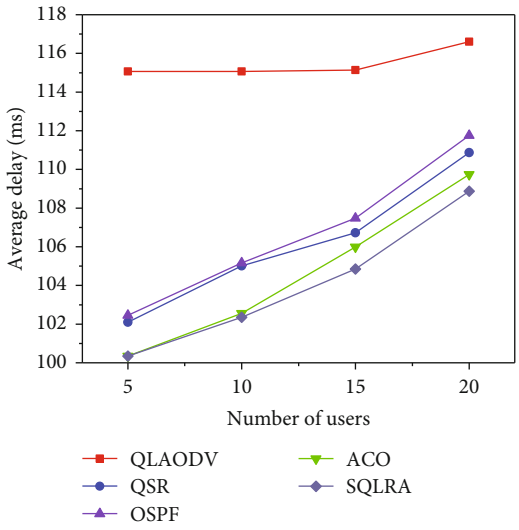


Figure 10: Average delay with varying number of users.

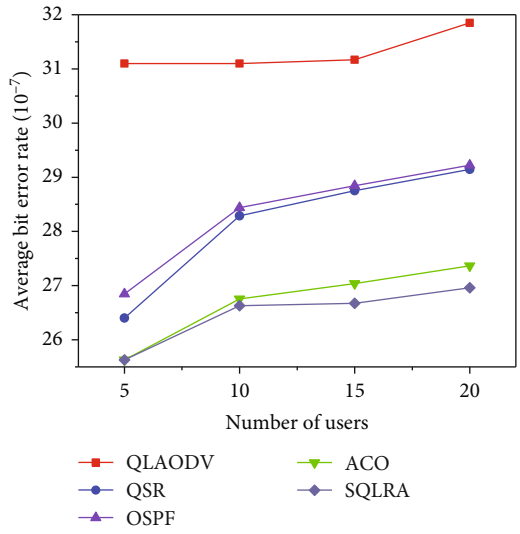


Figure 11: Average bit error rate with varying number of users.

increasing, the average delay obtained by all algorithms becomes larger. Moreover, we note that the curve of QLAODV grows slowly compared with other algorithms. For QLAODV, it prefers to select path with fewer hops. In addition, only when the current path is disconnected or saturated due to the consumption of users, QLAODV starts to select new path. Therefore, QLAODV does not choose new path frequently. This explains why the curve of QLAODV is mostly unchanged at the beginning. For QSR, OSPF, ACO, and SQLRA, they all consider the delay of links when calculating the cost function. Therefore, they prefer to choose the path with lower delay. As the number of users increases, the current paths cannot meet the needs of users, and these algorithms begin to select new paths. At this time, the delay of the selected paths is greater than that of the previously selected paths. This is the reason why the average delay of paths by QSR, OSPF, ACO, and SQLRA increases. Furthermore, we also observe that SQLRA performs better than ACO. The main reason is that ACO always achieves the suboptimal solution, and SQLRA mostly attains the global optimum.

(C) Average bit error rate analysis

Figure 11 plots the average bit error rate changes with varying number of users. As the number of users increases, the average bit error rate obtained by all algorithms presents an upward trend. Because these algorithms consider the bit error rate characteristic of the satellite links when selecting the path, the paths with the lower bit error rate are selected at first. As the resources of links are consumed by the increasing users, these algorithms begin to choose new paths which have larger bit error rate. Compared with ACO, SQLRA has the best performance. Even if the number of users is largest, the average bit error rate of SQLRA algorithm is lowest.

(D) Average visible time analysis

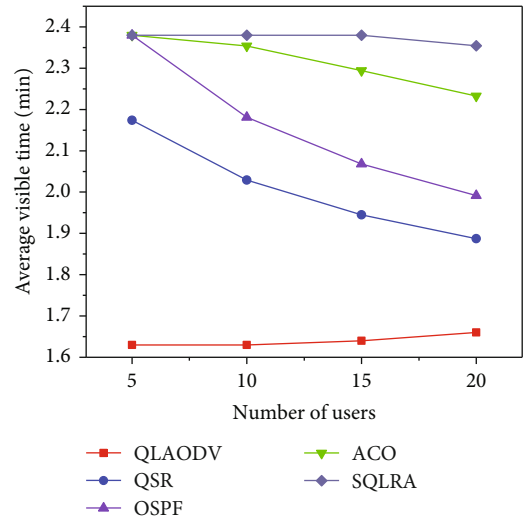


Figure 12: Average available time with varying number of users.

We show in Figure 12 the variation trend of the average visible time with varying number of users. We from Figure 12 observe that the average visible time of all algorithms shows a descend trend to varying degrees for algorithms QSR, OSPF, ACO, and SQLRA when the number of users is increasing. Considering that the visible time has an important impact on the performance of satellite network, these algorithms prefer to select paths with large visible time at the beginning. With more users accessing the network, the current paths cannot meet the requests of users. These algorithms start to select new paths with lower visible time. This explains why the average visible time of these algorithm gradually decreases. For QLAODV, the visible time of the links is not considered when selecting path. Therefore, when selecting the path, QLAODV algorithm does not prefer to choose the link with long visible time.

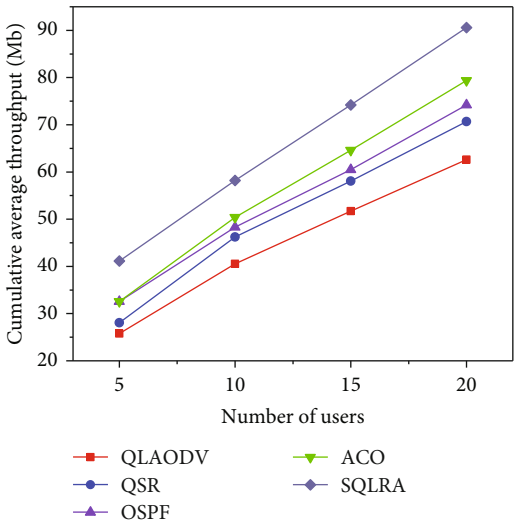


Figure 13: Cumulative average throughput with varying number of users.

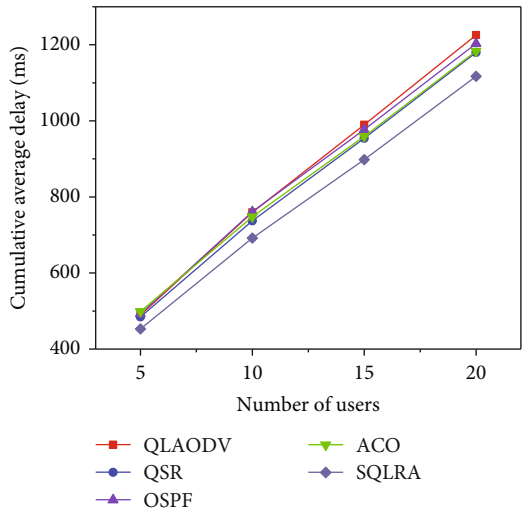


Figure 14: Cumulative average delay with varying number of users.

And this explains why the average visible time curve of QLAODV algorithm does not show a downward trend.

(2) Performance in communication scenario with different node pairs

The users' requests in this scenario have different source satellite nodes and destination satellite nodes. Considering the destination nodes of users are different, we use cumulative average throughput, cumulative average delay, cumulative average bit error rate, and cumulative average visible time as metrics, which can more clearly and reasonably evaluate the overall performance of the algorithms.

(A) Cumulative average throughput analysis

Figure 13 shows the relationship between the cumulative average throughput and the number of users. We note from Figure 13 that as the number of users increases, the cumulative average throughput of all algorithms presents an upward trend to varying degrees. Moreover, we also find that SQLRA has the best performance and QLAODV has the worst performance. The main reason is that when selecting the next hop, SQLRA considers the impact of the next link on the current link, and it attains the global optimum, while QLAODV does not consider the visible time of links when selecting the next hop. Furthermore, we also know that the visible time of the satellite link has a great impact on the network performance. When selecting the next hop, OSPF considers the available bandwidth, bit error rate, delay, and visible time. However, it adopts greedy strategy to select the next hop, which is easy to fall into local optimum. Compared with OSPF, ACO algorithm is initialized by random strategy, and it achieves global optimum as much as possible in the process of routing.

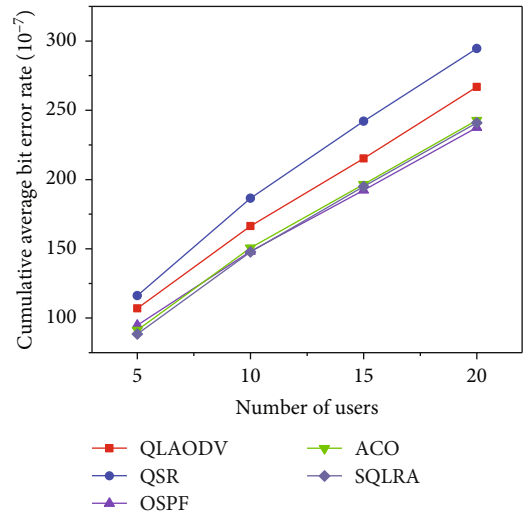


Figure 15: Cumulative average bit error rate with varying number of users.

(B) Cumulative average delay analysis

Figure 14 presents the change of cumulative average delay with varying number of users. As the number of users increases, the cumulative average delay obtained by all algorithms is increasing. Because users have different destinations, the paths selected by all algorithms are different. With the number of paths increasing, the total delay of the selected paths in the network increases. Therefore, the cumulative delay of the paths also increases. We note that SQLRA performs better than other algorithms. The main reason is that when the user's request is satisfied, SQLRA tends to choose the path with less delay, and it almost get the global optimum. We also observe that the performance of algorithms QSR, OSPF, and ACO is almost the same.

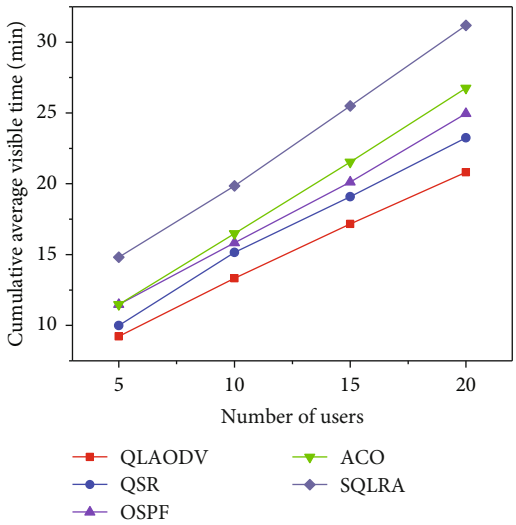


FIGURE 16: Cumulative average visible time with varying number of users.

The main reason is that although these algorithms all consider the delay of satellite links when computing the cost function, they use heuristic strategy to select the optimal path.

(C) Cumulative average bit error rate analysis

Figure 15 demonstrates the cumulative average bit error rate with varying number of users. We find from Figure 15 that the performance of QSR is worst and that of OSPF is best. We observe that although the performance of SQLRA is not the best, it is close to that of OSPF. Moreover, we also note that the performance of QLAODV is not worst compared with the curve in Figure 11. The main reason is that due to the paths of users are different, QLAODV algorithm begins to choose a new path when a new user arrives. This operation reduces the probability of selecting the path with high bit error rate. In addition, QLAODV algorithm considers the bit error rate when selecting new path.

(D) Cumulative average visible time analysis

Figure 16 shows cumulative average visible time with different number of users. From Figure 16, we see that as the number of users increases, the cumulative average visible time of all algorithms presents an upward trend to varying degrees. We also find that SQLRA performs better than other algorithms. When the number of users is 5, algorithms ACO and OSPF have the same performance. However, as the number of users increases, ACO performs better than OSPF. The main reason is that compared with ACO, OSPF is easier to fall into the local optimum. In the second scenario, the source nodes and destination nodes requested by users are different. As the number of paths increases, the total visible time of paths also increases. Therefore, the cumulative average visible time of the selected paths increases gradually.

By testing our proposed SQLRA algorithm in two different cases, we find that SQLRA performs better than other algorithms. In addition, we also find that SQLRA not only has good performance but also has strong robustness. In practice, we train the SQLRA algorithm at the ground control center, which reduces the consumption of computing resources and storage resources of satellites. Therefore, SQLRA algorithm is very suitable for dynamic satellite networks.

6. Conclusions

In this paper, we investigated the routing problem in STINs. We considered that selecting different satellite routes has an essential impact on the QoS of users and satellite network performance. We modelled the routing problem as a finite-state Markov decision process and proposed a routing algorithm based on Q-learning (QLRA). In addition, to solve the problem of slow convergence speed of QLRA algorithm, we proposed a split-based speed-up convergence strategy and designed a speed-up Q-learning based routing algorithm (SQLRA) and adopted a back-to-front update scheme to further improve the convergence speed of SQLRA algorithm. Moreover, we evaluated the performance of SQLRA in two different scenarios. Experimental results show that SQLRA algorithm has the best communication performance compared with other routing algorithms.

Although SQLRA algorithm performs better than other routing algorithms in two different scenarios, it does not have the ability of online learning. When the number of users in satellite network changes or some satellites do not work well, SQLRA needs to retrain model. In the future research, we are going to use deep neural network to design a routing algorithm with online learning ability. When the number of users changes or the link state of satellite network changes, SQLRA algorithm is able to update the existing model and reduce the training time as much as possible. In addition, traffic scheduling and satellite handoff management are also our future research issues.

Data Availability

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

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science Foundation of China (No. 61772385).

References

[1] F. Boccardi, R. W. Heath Jr., A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[2] H. Yao, L. Wang, X. Wang, Z. Lu, and Y. Liu, "The space-terrestrial integrated network: an overview," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 178–185, 2018.

[3] N. U. L. Hassan, C. Huang, C. Yuen, A. Ahmad, and Y. Zhang, "Dense small satellite networks for modern terrestrial communication systems: benefits, infrastructure, and technologies," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 96–103, 2020.

[4] J. P. Choi and C. Joo, "Challenges for efficient and seamless space-terrestrial heterogeneous networks," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 156–162, 2015.

[5] A. Guidotti, B. Evans, and M. Di Renzo, "Integrated satellite-terrestrial networks in future wireless systems," *International Journal of Satellite Communications and Networking*, vol. 37, no. 2, pp. 73–75, 2019.

[6] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: integrating terrestrial-satellite networks into 5G and beyond for data offloading," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 47–62, 2019.

[7] K. An and T. Liang, "Hybrid satellite-terrestrial relay networks with adaptive transmission," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 12448–12452, 2019.

[8] A. A. Dowhuszko, J. Fraire, M. Shaat, and A. Pérez-Neira, "LEO satellite constellations to offload optical terrestrial networks in placement of popular content in 5G edge nodes," in *2020 22nd International Conference on Transparent Optical Networks, International Conference on Transparent Optical Networks-ICTON*, pp. 1–6, Bari, Italy, 2020.

[9] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 756–767, 2002.

[10] C. Hedrick, "Routing information protocol," *Request for Comments*, vol. 1058, 1988.

[11] S. Anamalamudi, A. R. Sangi, M. Alkatheiri, and A. M. Ahmed, "AODV routing protocol for cognitive radio access based Internet of Things (IoT)," *Future Generation Computer Systems-the International Journal of Escience*, vol. 83, pp. 228–238, 2018.

[12] W. Liu, Y. Tao, and L. Liu, "Load-balancing routing algorithm based on segment routing for traffic return in LEO satellite networks," *IEEE Access*, vol. 7, pp. 112044–112053, 2019.

[13] X. Qi, B. Zhang, and Z. Qiu, "Joint rate control and load-balancing routing with QoS guarantee in LEO satellite networks," *IEICE Transactions on Communications*, vol. E103B, no. 12, pp. 1477–1489, 2020.

[14] Z. Na, Z. Pan, X. Liu, Z. Deng, Z. Gao, and Q. Guo, "Distributed routing strategy based on machine learning for LEO satellite network," *Wireless Communications & Mobile Computing*, vol. 2018, pp. 1–10, 2018.

[15] M. Majd and R. Safabakhsh, "Correlational convolutional LSTM for human action recognition," *Neurocomputing*, vol. 396, pp. 224–229, 2020.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[17] C. X. Jiang, H. J. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[18] N. Kato, Z. M. Fadlullah, B. Mao et al., "The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 146–153, 2017.

[19] F. Tang, B. Mao, Z. M. Fadlullah et al., "On removing routing protocol from future wireless networks: a real-time deep learning approach for intelligent traffic control," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 154–160, 2018.

[20] R. F. Dhila, T. M. Hamdani, and A. M. Alimi, "A multi objective particles swarm optimization algorithm for solving the routing pico-satellites problem," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1402–1407, Seoul, Korea, 2012.

[21] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2019.

[22] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.

[23] C. Jiang and X. Zhu, "Reinforcement learning based capacity management in multi-layer satellite networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4685–4699, 2020.

[24] V. Kirilin, A. Sundarrajan, S. Gorinsky, and R. K. Sitaraman, "RL-cache: learning-based cache admission for content delivery," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2372–2385, 2020.

[25] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco, and F. Liberal, "Virtual network function placement optimization with deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 292–303, 2020.

[26] J. Liu, R. Z. Luo, T. Huang, and C. W. Meng, "A load balancing routing strategy for LEO satellite network," *IEEE Access*, vol. 8, pp. 155136–155144, 2020.

[27] S. Geng, S. Liu, Z. Fang, and S. Gao, "An optimal delay routing algorithm considering delay variation in the LEO satellite communication network," *Computer Networks*, vol. 173, article 107166, 2020.

[28] S. Wei, H. Cheng, M. Liu, and M. Ren, "Optimal strategy routing in LEO satellite network based on cooperative game theory," in *International Conference on Space Information Network*, pp. 159–172, Singapore, 2017.

[29] Z. Jiang, C. Liu, S. He, C. Li, and Q. Lu, "A QoS routing strategy using fuzzy logic for NGEO satellite IP networks," *Wireless Networks*, vol. 24, no. 1, pp. 295–307, 2018.

[30] T. Pan, T. Huang, X. Li, Y. Chen, W. Xue, and Y. Liu, "OPSPF: orbit prediction shortest path first routing for resilient LEO satellite networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, 2019.

[31] L. Hao, P. Ren, and Q. Du, "Satellite QoS routing algorithm based on energy aware and load balancing," in *2020 12th International Conference on Wireless Communications and Signal Processing*, pp. 685–690, Wuhan, China, 2020.

[32] Z. Ji, S. Wu, C. Jiang, D. Hu, and W. Wang, "Energy-efficient data offloading for multi-cell satellite-terrestrial networks," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2265–2269, 2020.

[33] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2019.

[34] N. Torkzaban, A. Gholami, J. S. Baras, and C. Papagianni, "Joint satellite gateway placement and routing for integrated satellite-terrestrial networks," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, 2020.

[35] H. Xu, D. Li, M. Liu, G. Han, and C. Xu, "A hybrid routing algorithm in terrestrial-satellite integrated network," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 90–95, Chongqing, China, 2020.

[36] Q. Guo, R. Gu, T. Dong et al., "SDN-based end-to-end fragment-aware routing for elastic data flows in LEO satellite-terrestrial network," *IEEE Access*, vol. 7, pp. 396–410, 2019.

[37] Y. Liu, D. Lu, G. Zhang, J. Tian, and W. Xu, "Q-learning based content placement method for dynamic cloud content delivery networks," *IEEE Access*, vol. 7, pp. 66384–66394, 2019.

[38] S. Pan, P. Li, D. Zeng, S. Guo, and G. Hu, "A Q-learning based framework for congested link identification," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9668–9678, 2019.

[39] Z. Wei, F. Liu, Y. Zhang, J. Xu, J. Ji, and Z. Lyu, "A Q-learning algorithm for task scheduling based on improved SVM in wireless sensor networks," *Computer Networks*, vol. 161, pp. 138–149, 2019.

[40] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 247–257, 2020.

[41] M. Alhabo, L. Zhang, and N. Nawaz, "GRA-based handover for dense small cells heterogeneous networks," *IET Communications*, vol. 13, no. 13, pp. 1928–1935, 2019.

[42] C. Wu, K. Kumekawa, and T. Kato, "Distributed reinforcement learning approach for vehicular ad hoc networks," *IEICE Transactions on Communications*, vol. 93-B, no. 6, pp. 1431–1442, 2010.

[43] T. Li, H. Zhou, H. Luo, and S. Yu, "SERvICE: a software defined framework for integrated space-terrestrial satellite communication," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 703–716, 2018.

[44] H. Alayed, F. Dahan, T. Alfakih, H. Mathkour, and M. Arafah, "Enhancement of ant colony optimization for QoS-aware web service selection," *IEEE Access*, vol. 7, pp. 97041–97051, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5534270, 17 pages
https://doi.org/10.1155/2021/5534270

Research Article

User-Level Membership Inference for Federated Learning in Wireless Network Environment

Yanchao Zhao,[1] **Jiale Chen**,[1] **Jiale Zhang**,[2] **Zilu Yang**,[1] **Huawei Tu**,[3] **Hao Han**,[1] **Kun Zhu**,[1] **and Bing Chen**[1]

[1]*Collage of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China*
[2]*School of Information Engineering, Yangzhou University, China*
[3]*Computer Science & Information Technology, La Trobe University, Australia*

Correspondence should be addressed to Yanchao Zhao; yczhao@nuaa.edu.cn

Received 13 January 2021; Accepted 2 October 2021; Published 20 October 2021

Academic Editor: Jun Cai

With the rise of privacy concerns in traditional centralized machine learning services, federated learning, which incorporates multiple participants to train a global model across their localized training data, has lately received significant attention in both industry and academia. Bringing federated learning into a wireless network scenario is a great move. The combination of them inspires tremendous power and spawns a number of promising applications. Recent researches reveal the inherent vulnerabilities of the various learning modes for the membership inference attacks that the adversary could infer whether a given data record belongs to the model's training set. Although the state-of-the-art techniques could successfully deduce the membership information from the centralized machine learning models, it is still challenging to infer the member data at a more confined level, the *user level*. It is exciting that the common wireless monitor technique in the wireless network environment just provides a good ground for fine-grained membership inference. In this paper, we novelly propose and define a concept of user-level inference attack in federated learning. Specifically, we first give a comprehensive analysis of active and targeted membership inference attacks in the context of federated learning. Then, by considering a more complicated scenario that the adversary can only passively observe the updating models from different iterations, we incorporate the generative adversarial networks into our method, which can enrich the training set for the final membership inference model. In the end, we comprehensively research and implement inferences launched by adversaries of different roles, which makes the attack scenario complete and realistic. The extensive experimental results demonstrate the effectiveness of our proposed attacking approach in the case of single label and multilabel.

1. Introduction

With the revolution of decentralized machine learning, researches on collaborative learning technologies such as federated learning for resource-constrained devices on mobile edge networks [1] have been increasing and expanding the landscape of use cases. Federated learning [2] enables mobile devices to collaboratively learn a shared prediction model while keeping all the training data locally instead of in the cloud, which may be at risk of privacy leakage. Unlike other collaborative learning frameworks, federated learning updates a global model by aggregating all local parameters from participants, so that the federated model can benefit from a wide range of non-IID [3] and unbalanced data distribution among diverse participants. In keeping with the vigorous development of 5G network technology, as demonstrated in Figure 1, federated learning empowered by wireless networks stimulates the potential of some applications around us, making them smarter and more powerful. Furthermore, edge wireless networks for content caching and data calculation are a promising way to reduce backhaul traffic load. Federated learning, based on a model that uses local training

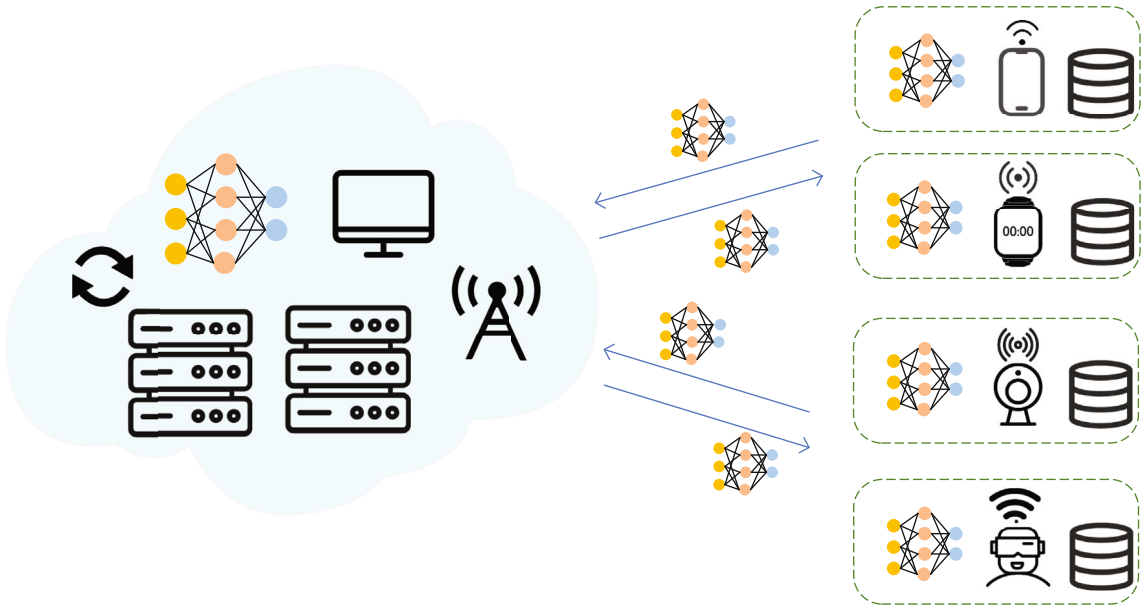


Figure 1: Federated learning for mobile networks.

instead of direct access to participants' data, seems like a perfect fit for content popularity prediction in proactive caching in wireless networks [4].

Although federated learning can provide a basic privacy guarantee with localized training, privacy issues still exist during the aggregation and communication process. Emerging attacking methods, including membership inference, have been undermining the security of federated learning and even the entire artificial intelligence technology. Basically, the membership inference problem is a classification problem that the adversary needs to tell whether the data with unknown ownership is part of a certain collection or not. Although this is an indirect privacy stealing, when membership inference attacks are used as preattacks for other attack scenarios, such as the reconstruction attack [5], the membership information makes these attacks more targeted and disruptive. Shokri et al. [6] first proposed the membership inference against a black-box machine learning API. In this case, the adversary can construct a "shadow model" by obtaining the fluctuation difference in the confidence of the black-box output of similar data obtained from the target model (e.g., "MLaaS" platform), thereby approximating the behavior of the target model to spy the privacy of the training set. In this way, the adversary does not need to get the internal structure and parameters of the target model. However, this attack has many assumptions that the adversary can directly call the API, which is equivalent to the adversary's unauthorized access to the model, and use it in a centralized learning environment. Moreover, the adversary has a dataset from the same distribution as the target model's training data.

For the recent researches on the security issues of artificial intelligence, Salem et al. [7] improved Shokri et al.'s method by containing multiple neural network models in a stack, which is sensitive to the membership information. In this way, the attack model can only focus on the relationship between the membership information and the classification results, even if the data is from different distributions. Nasr et al. [8] proposed a membership inference attack launched from the participant side. The core technique of the scheme was the stochastic gradient ascent (SGA). The adversary extracted the parameters of the target model during the training process, including gradients and loss rates, into fully connected layers to train the neural network. When the gradients of data are forced to increase by SGA every time, the gradients of member data will be forcibly decreased by the stochastic gradient descent (SGD) [9], while the gradients of nonmember data still rise. By detecting this distinction, the membership information is transformed into a score, which is used as a new feature to construct unsupervised learning to distinguish member data from nonmember data.

However, although the above inference attacks can reveal the privacy of training data to varying degrees, there are some limitations when they are transplanted to federated learning. Firstly, in the previous centralized learning, the dataset used to train the attack model had the same distribution as the dataset belonging to the target model, and even these datasets have a certain proportion of intersection. Secondly, there is no research on the possible existence of malicious participants launching the membership inference, which is divorced from the real situation. Usually, the prerequisite for the successful operation of federated learning is that all participants and servers are honest, but as long as there are malicious members in the cluster, such as doing poisoning attacks, this perfect mechanism will be out of balance. After all, no one can guarantee that federated learning is always perfect. Thirdly, even if an attack is to be launched from the client side, the inherent privacy protection mechanism of federated learning, aggregated algorithm, will prevent it from succeeding. Fortunately, in a wireless environment, with a wireless monitor mode, the situation has changed a lot. Motivated by the function of the wireless

monitor and those shortcomings in existing inference approaches, we give a deep analysis of active and targeted membership inference attacks in the federated learning with a white-box access model. We are established on the wireless network environment to implement all the elements from the perspective of a malicious participant and a malicious central server. We name our scheme as the *user-level* membership inference. The reason why we call it "*user-level*" is that we have refined the target of inference from the previous global model to a certain participant (*victim*), caring more about his membership privacy. We try to play two roles of malicious participant and server, respectively, in the federated learning mode to launch inference (see Figure 2). Based on the traditional membership inference in centralized and distributed learning, we take a more practical threat assumption that the adversary does not need to know any prior knowledge about the training datasets. Stuck by the model averaging algorithm and the lack of training data for the membership inference, we make full use of the characteristics of the wireless monitor to further propose a local-deployed data augmentation method relying on the generative adversarial networks (GANs) to generate high-quality fake samples.

Our contributions in this paper can be summarized as follows.

(i) *User-Level Membership Inference*. We further disclose the security hole of the current federated learning enabled by 5G wireless networks with novelly launching fine-grained membership inference attacks and encouraging more researches on preventing participants from leaking privacy.

(ii) *Data Augment Using GANs*. To gain insight into the data distribution of other participants to perform the membership inference, we use the information obtained by a wireless monitor and innovatively develop local-deployed generative adversarial networks (GANs) to generate samples with all labels.

(iii) *Systematic Analysis in All Positions*. We progressively launch membership inference from a malicious participant in federated learning. In addition, we investigate and validate the possibility of launching inferences from the server side.

(iv) *Excellent Performances in Experiment*. In experiments, we set two major indexes, the accuracy of the membership inference and the learning task, to measure the effectiveness of our scheme. We also performed multiple sets of comparative experiments to prove the impact of the number of labels on the membership inference attack.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 briefs some background knowledge and introduces the threat model. Section 4 describes the proposed method framework along with an analysis of the membership inference. The performance evaluation results are presented in Section 5. Section 6 discusses the limitations of our method and gives some ideas. Finally, Section 7 concludes this paper.

2. Related Work

In this section, we will introduce the privacy protection methods for distributed deep learning and federated learning. After that, we will refine the issue of privacy leakage to the membership inference attack. Finally, we present the various attacks against a specific victim in the federated learning scenario.

2.1. Privacy-Preserving Distributed Learning and Federated Learning. The traditional centralized machine learning, where the data holder trains the model locally, is limited by the computing resources and data volume. It is difficult to meet the current needs for massive data calculations, data diversity, and storage performance. As a result, the distributed learning framework emerges, providing a collaborative training scenario. But once the third party involves, there will be a problem of privacy leakage. To protect distributed learning, an algorithm named as distributed selective stochastic gradient descent (DSSGD) was proposed by Shokri and Shmatikov [10]. The results showed that even if only 1% of the parameters are shared, collaborative learning will bring a higher accuracy than centralized learning. Moreover, Shokri and Shmatikov [10] utilized differential privacy [11] to effectively prevent data privacy that may be indirectly leaked. Based on the previous article, Phong et al. [12] proposed four cases of indirect privacy leakage and pointed out that even if some gradients are uploaded randomly, there are still significant hidden privacy risks. The author introduced homomorphic encryption technology [13] in the large-scale distributed neural network to ensure that the cloud server cannot steal the privacy of data during the entire process of model training. The only drawback was computationally expensive and time-consuming. Zhang et al. [14] present BatchCrypt, which is a system solution for cross-silo federated learning. In the scheme, they encode a batch of quantized gradients into a long integer and encrypt it in one go, instead of encrypting individual gradients with full precision. BatchCrypt substantially reduces the communication overhead caused by homomorphic encryption.

The difference between collaborative learning and federated learning is that the central server of federated learning will average the updates (i.e., the weight matrix) after each communication round. Even so, the privacy violation remains a challenge. In the user-level differential privacy algorithm proposed by [15], this average is changed and approximated using a random mechanism. This is done to hide the contributions of individual participants in the collection, thereby protecting the entire distributed learning process. Truex et al. [16], in order to compensate for the impact of differential privacy on model accuracy, combining differential privacy with secure multiparty calculations, reduced the noise injection caused by the increase in the number of participants and maintained the accuracy and privacy of the model. Inspired by these efforts, we began to focus on privacy preserving in federated learning.

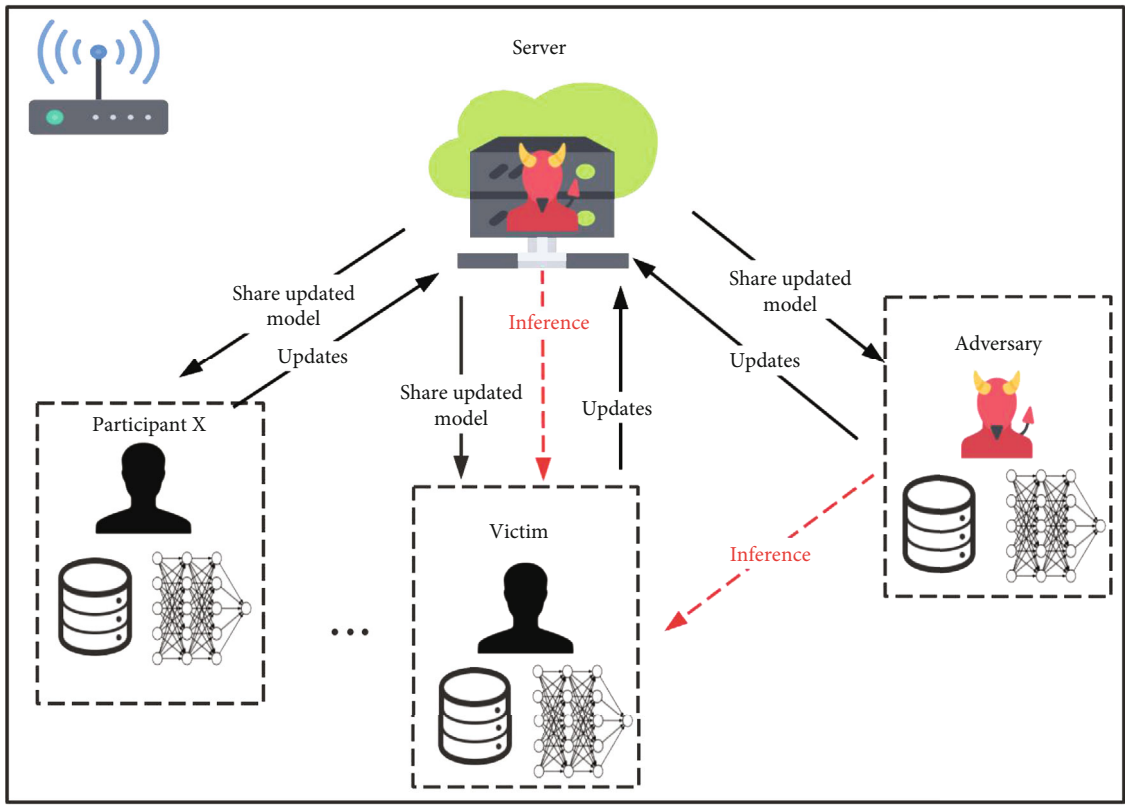


Figure 2: Multiple membership inference methods in federated learning under wireless network.

2.2. Membership Inference Attack. The membership inference attack means that when a record is given to the inference model, the model can tell whether the record belongs to a target's training set. As centralized learning evolves to distributed learning, there are many variants of the membership inference, which can be divided into active attacks and passive attacks, including those launched by a malicious server and by malicious participants [8]. Not surprisingly, the more participants are involved, the less information that adversary can learn from another participant. In other words, the accuracy of the membership inference attack will decrease as the number of participants increases. Taking into the situation of numerous participants, the active local adversaries are facing challenges of lacking training data. Besides, the research found that even a model with differential privacy protection still has the risk of leaking membership privacy [17]. The main reason for the leakage of membership information is model overfitting [18–20]. At present, membership inference has become the third largest attack against AI systems, accounting for 3.5%. Our work focuses on membership inference in federated learning. But what needs to be distinguished from the membership inference mode under centralized learning is that the goal of our proposed method under federated learning is to infringe the privacy of a certain participant, not the privacy of the entire training dataset. We summarize this more fine-grained analysis as the membership inference attack under federated learning. This takes the membership inference a step further in the field of study.

2.3. Attacks against a Specific Victim in the Collaborative and Federated Learning. In addition to stealing membership information, there are many attacks on a certain participant in federated or centralized learning. These attacks, for example, the poisoning attack [21], the model inversion attack [22], the representative inference [23], the model stealing attack [24], and the capturing of extra properties [25], mainly assume that the adversary, whether a malicious server or a malicious participant, actively launches attacks and tries to induce the victim to output more private information to achieve the purpose. However, attacks from the client side in federated learning are limited to recovering class-wised representatives rather than mining user-level privacy because the malicious participant can only access updates aggregated by the server (contributed by all the participants). Therefore, to launch these attacks, more auxiliary information is often required, e.g., class labels or other participant-wise properties. Our method cuts into the crux of this nodus from the wireless monitor technique and the generative adversarial networks (GANs).

3. Preliminaries

In this section, we introduce the background knowledge and the other preliminaries, including assumptions and the threat model of our method.

3.1. Federated Learning. Federated learning [26] is a distributed deep learning solution first proposed by Google in

2016. In the selection phase of the federated learning, the server will randomly and partly select participants to participate in this round of training. In the reporting phase, the server will wait for each participant to return the trained gradient parameters. After the server receiving parameters, it will use an algorithm to aggregate them and notify participants of the next request time. If there are enough participants returning gradients before the timeout, this round of training is successful; otherwise, it fails. In the entire system, there is a pace control module (Pace Steering), which can manage the connection of all the participants. For the small-scale federated learning training, Pace Steering guarantees that sufficient participants are involved in each round of training. For the large-scale federated learning training, Pace Steering will randomize the request time of the participants to avoid a large number of simultaneous requests, which may cause problems. By the way, the models trained by each participant do not interfere with each other during the training process.

In 2017, Google's Mcmahan et al. proposed the FedAvg algorithm, which is a synchronous protocol [27]. The updates are averaged and accumulated to the current shared model. Equation (1) demonstrates the process. M_t denotes the shared model at the tth iteration, M_{t+1} means the newest model, and u_t^k indicates the update from the kth client at iteration t:

$$M_{t+1} = M_t + \frac{1}{N}\sum_{k=1}^{N} u_t^k. \tag{1}$$

All participants execute Equation (2) in each epoch, where η is the learning rate and b means the batch. Finally, every participant returns his w, weights, to the server:

$$W = W - \eta \nabla L(W; b). \tag{2}$$

On the one hand, federated learning can effectively enrich the diversity of training sets and allow more data to participate in calculations. On the other hand, federated learning allows data to be stored locally, which meets some data-sensitive requirements, such as medical and military scenarios. But this does not mean that privacy will not be a problem in federated learning. Inference against a certain participant's data and output greatly threatens the security of the federated learning.

3.2. Wireless Monitor Mode. By default, the wireless network card and the wireless access point (WAP) [28] are in a managed mode after establishing a connection. In this mode, the wireless network card only serves to receive data sent to itself from the WAP. If you want the wireless network card to monitor all communication information in the wireless environment, you can set the wireless network card to monitor mode (also called RFMON mode [29]) and then use software such as Wireshark to capture data packets for analysis, as shown in Figure 3. Under the normal setting, the network card will only accept data packets sent to itself and discard all other packets. Of course, the network card can accept all the messages unconditionally; this is the so-called promiscuous mode. However, unlike the promiscuous mode, the monitor mode does not require a connection to a WAP or ad hoc network [30]. The monitor mode is special unique to the wireless network card, and the promiscuous mode is applied to the wired and wireless network cards. With the help of the wireless monitor mode, it is convenient for us to collect auxiliary information for the client-side's membership inference.

3.3. Generative Adversarial Networks. Generative adversarial nets (GANs) were first proposed by Mirza and Osindero [31], which is a neural network trained in an adversarial manner. GANs contain two competing neural network models. One is a generator G that draws random samples z from a prior distribution (e.g., Gaussian or uniform distribution) as the inputs, and then, G generates samples from z that approximate the input distribution. Another model is a discriminator D. Given a training set, the discriminator D is trained to distinguish the generated samples from the training (real) samples. Equation (3) shows the objective function of GANs. Briefly speaking, GANs are trained to minimize the divergence between the generated and real data distribution.

$$\min_G \max_D V(D, G) = \mathscr{E}_{x\sim P_{\text{data}}(x)}[\log(D(x))] + \mathscr{E}_{z\sim P_z(z)}[\log(1 - D(z))], \tag{3}$$

$$V(D, G) = \int_x P_{\text{data}}(x) \log(D(x)) + P_g(x) \log(1 - D(x)) dx, \tag{4}$$

$$D(x) = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_g(x)}. \tag{5}$$

The P_{data} and P_z denote the training (real) distribution and prior distribution, respectively. P_g indicates the distribution of generated samples. These two models G and D are trained alternately until this minimax game achieves Nash equilibrium [32], where the generated samples are difficult to be discriminated from the real ones. The proof of Equation (4) shows that there is an optimal discrimination model. According to the nature of JS divergence [33], if and only if the generation distribution P_g is equal to the real data distribution P_{data}, $JSD(P_{\text{data}} \| P_g) = 0$, we can obtain the optimal discriminator, as shown in Equation (5). It should be noted that the GANs originally set by Mirza and Osindero [31] were defined only in the real number (continuous data, such as images), because the output gradient of the discriminator D would give feedback on how to make it more realistic by slightly changing the generated data. However, when it comes to discrete data (e.g., text), the variation among tokens is much greater than that of images. In addition, the GANs can only evaluate the entire sequence, not the quality and trend of the partial sequence.

Committed to overcoming this problem, the new idea is to introduce reinforcement learning (e.g., SeqGAN [34]). The generator is an agent, the state is the generated token,

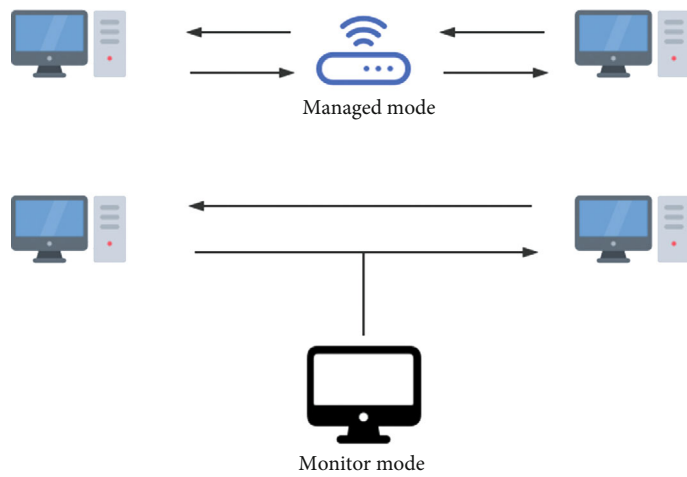


Figure 3: Two wireless card modes.

and the action is the next generated token. Monte Carlo Tree Search (MCTS [35]) is used to estimate the state behavior value and complete the various possibilities of each action. D generates a reward for these complete sequences, as shown in Equation (6), passes it back to G, and updates G through reinforcement learning to train generative networks that can produce the next optimal action. The completed Q function is shown in Equation (7):

$$Q_{D\phi}^{G_\theta} = \left(a = y_T, s = Y_{1:T-1} = D_\phi\left(Y_{1:T}\right)\right), \tag{6}$$

$$Q_{D\phi}^{G_\theta} = (s = Y_{1:t-1}, a = yt) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n), Y_{1:T}^n \in MC^{G_\beta}(Y_{1:t}; N), & t < T, \\ D_\phi(Y_{1:t}), & t = T. \end{cases} \tag{7}$$

Unfortunately, the experimental results prove that this idea does work, and the loss of synthetic data does decrease, but the quality of the generated text is very poor on the real Chinese poetry, which directly affects how this batch of data should be labeled and then becomes the training data of the membership inference attack model. This is currently a bottleneck. The work of Fedus et al., MaskGAN [36], may point out a direction for improving the quality of the generated text.

3.4. Assumptions. As done in previous works, participants will declare the labels of the local data before they start training, which can be verified through a wireless monitor. In fact, this behavior does not reveal valuable privacy about the training set. Because the label cannot reflect the attributes of the data, our scheme is based on a preliminary assumption that the sample labels owned by participants do not overlap. Taking the MNIST dataset as an example, we assume that participant P_1 has data samples with labels "0," "1," and "5"; participant P_2 has data samples with labels "2" and "7"; and so on. Under these circumstances, the declared label "1" cannot reflect the attribute of digit "1" in the picture, such as whether the font is inclined to the right or the left. The purpose of this nonintersecting setting is to facilitate the attack model to compare the results of the attack with the previously declared information to implement the membership inference attack. For example, in training medical data, in order to enrich the training set, different hospitals label the data according to their different pathological information. In this way, the federated model can obtain more pathological classes. Of course, there should be samples with the same label between different hospitals. The membership inference in this case will be discussed in Section 6.

3.5. Threat Model. Here, we will elaborate on the conditions that the adversary has.

3.5.1. Adversary's Objectives. In our settings, the ultimate objective of the adversary is to obtain indirect information about the target victim's dataset. So, we set two indexes in the context of classification tasks to evaluate our attack model: (1) *membership inference accuracy*: means the classification confidence of the target dataset; (2) *main task accuracy*: denotes that the global model should maintain a high prediction accuracy without overfitting.

3.5.2. Adversary's Observations. What the adversary can observe depends on which role he plays. When the adversary is a malicious participant, he can obtain the aggregated white-box global model that is fed back from the central server after each iteration. Besides, with the help of the wireless monitor mode, the adversary can see the communication of other participants. And if we set the adversary as a malicious server, then he can clearly know which participant a certain parameter comes from at this round. (1) For the adversary from the client side, a white-box model and monitored information are sufficient to launch the inference attack. Since the honest server distributes updated models to various participants during each iteration, the adversary will keep the latest model snapshot with him. Therefore, everything of the global model is exposed to the adversary, such as the structure of the model, the algorithm L, and

the parameters θ of multiple versions. This is beneficial for us to use GANs to launch the membership inference attack. (2) For the adversary from the server side, the acquisition of target parameters is also divided into active acquisition and passive acquisition. Referring to the work of Wang et al. [23], the adversary can actively obtain the parameters of the victim and then use GAN for data recovery. The details of our proposed scheme will be introduced in the next section.

3.5.3. Adversary's Capabilities. This topic is also divided into two roles to elaborate. We will list what the adversary can do and cannot do to assess his capabilities. As a malicious participant, on the one hand, the adversary *can* (1) have a snapshot of each updated model, (2) fully control his local data and training procedure, (3) arbitrarily modify the hyperparameters, (4) randomly select local parameter updates over communication rounds, and (5) silently monitor the communication content of the entire wireless network environment. On the other hand, the adversary *cannot* (1) directly access other participants' local data. From the perspective of a malicious server, the adversary *can* (1) proactively obtain parameters uploaded by the victim and unearth useful information from them, (2) aggregate all the collected parameters, and (3) optionally feedback special models to the victim. Besides, the adversary *cannot* (1) see the data owned by participants diametrically.

4. Proposed Membership Inference Attack

In this section, we describe the detail of the user-level membership inference attack in federated learning. Specifically, we focus on some malicious situations in federated learning where a participant and the central server are separately considered an insider, who will silently bypass the attention of others in the cluster to complete the task of differentiating the record's ownership.

4.1. Malicious Participant's Perspective

4.1.1. Attack Overview. Figure 4 overviews the first approach we designed: participant's inference attack. This is an active attack by the adversary. We suppose that in a wireless local area network (WLAN) environment, there are N participants, where the victim V is the target participant, and the adversary A is also on the client side. In the kth iteration, both A and V download the same parameters θ_d. V normally uses parameters to update the local training model, then performs training, and finally returns the training update θ_u to the server. Since the server could honestly average the parameters received from various participants before updating the global model, it is hard for the adversary to explicitly get clues of the target victim to launch the membership inference even with the wireless monitor mode. Therefore, we take GANs as a tool for attack. Except using parameters for local training, A will also copy the parameter θ_d to discriminator D in GANs for updating synchronously, so that the generator G can continuously generate samples closer to the real samples. These generated samples will be used to train the ultimate attack model with the corresponding classification algorithm. When the target dataset is obtained, the attack model will predict results. If a sample whose prediction result is consistent with the declaration information, we can judge it as "IN"; otherwise, judge it as "OUT."

4.1.2. Reconstruction Data with GANs. The goal of our data augmentation phase is to make the training set for the attack model complete. The structure of GANs and details of the data augmentation phase are shown in Figure 5. The generative network $g(z;\theta_G)$ is initialized and generates data records from random noise. In the discriminative network $f(x;\theta_D)$, the discriminator D is initialized with the global model. In this way, replacing the network parameters of D with global model parameters is equivalent to training D directly on the overall training data. Let x_i be the original image in the training set and x_{gen} be generated images. We apply the optimization algorithm based on the approach proposed by Mirza and Osindero [31] and formulate the problem as

$$\min_{\theta_G}\max_{\theta_D}\sum_{i=1}^{n_+}\log\left(f(x_i;\theta_D)\right)+\sum_{i=1}^{n_-}\log\left(1-f\left(g\left(x_{\text{gen}};\theta_G\right);\theta_D\right)\right), \tag{8}$$

$$\mathscr{L}_G\left(\theta_g\right)=\mathbb{E}_{z\sim p(z)}[\log\left(D(G(z))\right)]. \tag{9}$$

The generator G wants to generate samples x_m of class m, which belongs to one of the training sets. G yields x_{gen} to discriminator D. If D can classify x_{gen} as class m, then the data augmentation phase sets $x_m \longleftarrow x_{\text{gen}}$ and returns x_m. Otherwise, it will update the generator G to minimize its loss $\mathscr{L}_G(\theta_g)$ as shown in Equation (9). The pseudocode of the data augmentation phase is shown in Algorithm 1. We first initialize the generation network G and use the current federated learning model as the discriminator D to calculate the gradient to distinguish the generated data from the original data. Until that the discriminator is unable to distinguish the generated data, we get the eligible generated samples x_{gen}.

4.1.3. Attack Algorithm. The pseudocode of the attack phase is shown in Algorithm 1. After generating samples with all labels, we begin to train a classification model. The selection of the inference algorithm can be determined after analyzing the specific generated samples as we described in Section 6. In our experimental scenario, we take the MNIST dataset as an example, and we use the CNN model accordingly. After the model training is completed, the adversary launches the membership inference attack against a bunch of data, named target dataset, which contains the training data of the victim and other participants. After the attack is over, we compare the prediction result with the label information declared by the victim. The data with the same comparison result is regarded as the victim's training data, marked as "IN." Other data, which has different comparison results, are marked as "OUT." To calculate the accuracy of our membership inference attack, we divide the number of

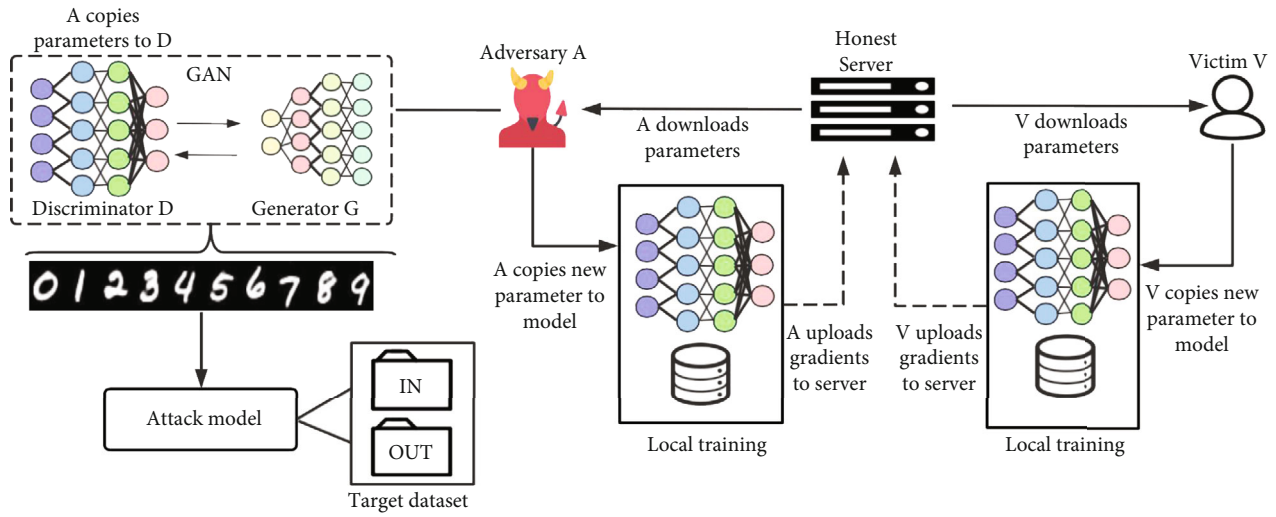


Figure 4: Overview of membership inference from the client side based on GANs.

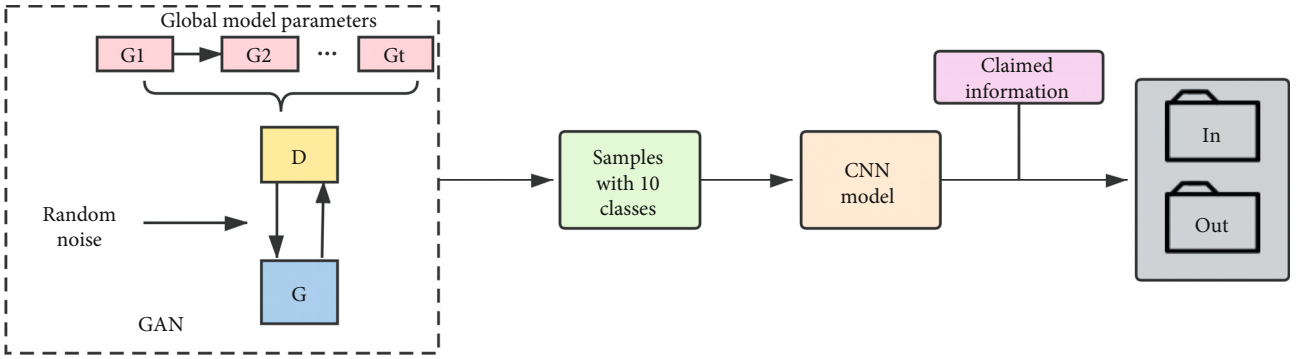


Figure 5: Data augmentation phase.

Input: The GANs iteration round $i_{\max}$, the federated learning model $f()$, generator G, discriminator D.
Output: The generated dataset $Data^{gen}$:(x,y) and the inference result '*IN*' or '*OUT*'.
1: **Procedure** Adversary Execution.
2: Initialize G
3: Set $D \longleftarrow f()$
4: **for** $(i = 1; i <= i_{\max}; i++)$ **do**
5: Run G to generate sample x_{gen}
6: Update G based on Eq (8)
7: **end for**
8: $y = f(x_{\text{gen}})$
9: Output: $D^{gen} : (x, y)$
10: $D^{train}_{attack} = D^{gen} : (x, y)$
11: Attack Phase:
12: Train CNN model using D^{train}_{attack} dataset.
13: Perform membership inference attack against D^{target}_{attack} dataset.
14: Compare the inference results with the claimed information.
15: Output: Mark every record as '*IN*' or '*OUT*', where '*IN*'represents the **Victim's** training sample.

Algorithm 1: Participant's attack procedure.

data marked "IN" by the number of victim's data in the target dataset. Detailed experimental results are presented in the next section.

4.2. Malicious Server's Perspective

4.2.1. Attack Overview. Figure 6 shows the second attack architecture: the server's inference attack. It should be emphasized that this is still an active attack from the server side. The so-called active denotes that the server is highly proactive, e.g., without undermining federated learning, the malicious server can cunningly employ unfair tricks to accomplish the purpose. We also assume that there are a total of N participants, including the victim V and the adversary A at this time on the server side. To achieve a better inference accuracy, from the beginning, the malicious server isolates the victim from other participants [8]. In this situation, the adversary can control the victim individually by sending the victim a quarantined version of the shared model, M_{iso}. Naturally, the adversary intentionally obtains valuable parameter information for reconstructing the victim's data. We adopt a similar approach to that used by Wang et al. [23], using an affiliated server that the victim V will connect to it without his knowledge. So that the shared model of each communication round between them is M_{iso} instead of M. There are two points to note: (1) with this setting, M_{iso} is still sharing weights with D in local-deployed GANs; (2) M_{iso}'s aggregation update is not synchronized with M, which means the target participant's model does not get aggregate with the parameters of other parties. It is aimed to let more membership privacy information be stored in the victim's model.

The next step is to train the membership-oriented GANs on the affiliated server, where the training process is the same as the normal GANs. But since the malicious server simply trains on the real data of the target victim at this time, the generated samples X_{gen} are of higher quality and easier to identify. When participants have multiple labels, but there is no label overlap between them, we can divide the data of these labels into "IN" according to the labels of X_{gen}. After acquiring a new batch of data, we can determine the membership.

In summary, while the server is doing evil, as we described in Section 3.5, the adversary's ability does not exceed the limit of the server's own ability. A lot of previous work has realized this situation.

4.2.2. Reconstruction Data with GANs in Affiliated Server. In the work of Phong et al. [37], the participant's data is restored based on the weight. The malicious server can access the updated parameters under federated learning. But the disadvantage of this scheme is that the shared model has to be a fully connected network. Besides, the update is needed to be obtained by training on a single sample. It is apparent that these constraints obstruct the applicability of the scheme to other algorithm models. Inspired by this solution, we break through the limitations of the shared model by introducing GANs deployed on the affiliated server to secretly access the victim's data privacy. In this attack mode, the operation of GANs is basically the same as that described above. The white-box model of the target victim will serve as the discriminator of GANs, gradually promoting the generator to output high-quality samples with the target labels.

4.2.3. Attack Algorithm. As the pseudocode described in Algorithm 2, at the beginning of federated learning, M_{iso} is initialized consistently with M, but these two models gradually become asynchronous. The affiliated server simply copies the parameters uploaded by the victim for use with GANs, so it will not interfere with the global model's training and will mislead the victim into thinking that he is contributing to global training. After the whole federated learning period, a fully functional global model held by the malicious server can be utilized to identify the generated samples and obtain labels, y_{gen}. The complete generated data with labels, $X_{\text{gen}} : (x_{\text{gen}}, y_{\text{gen}})$, are then marked as "IN," and the rest are marked as "OUT." In this way, they are supplied to build the attack model. The experimental results are also detailed in the next section.

5. Performance Evaluation

In this section, we evaluate our proposed methods, including GANs and the membership inference, in different ways.

5.1. Datasets and Evaluation Goals. In the participant inference experiment, we use two datasets, MNIST [38] and CIFAR-10, to verify the indicators of the program. Then, we put the AT&T dataset [39] into an application to test the performance of GANs in the server inference experiment. Our prepared data are mainly images because we believe that image data has a very special property; that is, the label of the image cannot fully reflect the content of the image. From a privacy perspective, this should be a loophole in the data used for training. Text data has also been considered to be introduced to enrich the experiment, but GANs did not achieve surprising results in natural language processing (NLP) tasks, which hindered the application of text data in our scheme. Many reasons are discussed above. Details of experimental datasets are described in Table 1.

(i) *MNIST*. This dataset includes ten classes of handwritten digits from "0" to "9," which is widely used for training and testing in the field of machine learning. It is commonly used in training various image processing models. A total of 70,000 images is divided into the training set (60,000 images) and the testing set (10,000 images). The grayscale image is normalized into 28×28, a total of 724 pixels.

(ii) *CIFAR-10*. It consists of a training set of 60,000 images and a testing set of 10,000 images with 32×32 pixies in ten classes. These images are mainly cats, dogs, horses, etc.

(iii) *AT&T*. The AT&T dataset (a.k.a. Olivetti dataset of faces) contains 40 topics, each with 10 pictures, and a total of 400 samples. The subject is Olivetti employees or Cambridge University students. The

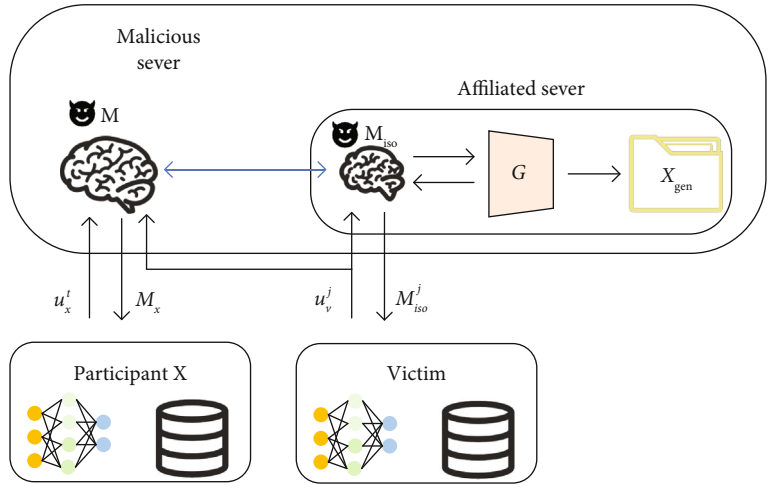


Figure 6: Overview of server-side membership inference using GANs on the affiliated server.

Input: The iteration round of GANs deployed on affiliated server i_{iso}, the federated learning model $f()$, The iteration round of normal training i_{normal}, generator G, discriminator D, isolated model M_{iso} and victim's updated parameters u_v.
Output: The generated dataset X_{gen} :(x,y) and the inference result '*IN*' or '*OUT*'.
Procedure Adversary Execution.
2: Initialize G
Deceptively connect the affiliated server with M_{iso} to the victim
4: Set $D \longleftarrow M_{iso} \longleftarrow M$
for $(i = 1; i <= i_{normal}; i++)$ **do**
6: Train global model $f()$ by collecting all parameters from all participants
Synchronize the following 'for' loop
8: Update $f()$ based on Eq (1)
end for
10: **for** $(i = 1; i <= i_{iso}; i++)$ **do**
Copy u_v to affiliated server
12: Run G to generate sample x_{gen} in a targeted manner
Update G based on Eq (8)
14: **end for**
$y = f(x_{gen})$
16:$y_{other} = f(x_{other})$
Output: $X_{gen} : (x_{gen}, y_{gen}) \longrightarrow$ IN
18: Output: $X_{other} : (x_{other}, y_{other}) \longrightarrow$ OUT
$D_{attack}^{train} = X_{gen} : (x, y) + X_{other} : (x, y)$
20: Attack Phase:
Train CNN model using D_{attack}^{train} dataset.
22: Perform membership inference attack against D_{attack}^{target} dataset.
Compare the inference results with the claimed information.
24: Output: Mark every record as '*IN*' or '*OUT*', where '*IN*'represents the **Victim's** training sample.

Algorithm 2: Server's attack procedure.

age of the subjects was concentrated between 20 and 35 years old. Among them, 36 are males and 4 are females. The subject's picture allows only limited lateral movement and limited tilt. So for most objects, the images are taken at different times and under different lighting conditions. The image was manually cropped and rescaled to 92×112 resolution, 8-bit gray levels. Five images of each object are used for training, and five images are used for testing, a total of 200 training images and 200 test images.

To comprehensively illustrate our proposed attack model, we set the following two goals: (1) mimic data generation: means the effectiveness of our proposed data augmentation algorithm and data reconstruction in server side using

Table 1: Summary of datasets used in our experiments.

Dataset	Labels	Input size	Training samples	Testing samples
MNIST	10	28 ∗ 28 ∗ 1	60000	10000
CIFAR-10	10	32 ∗ 32 ∗ 3	50000	10000
AT&T	40	92 ∗ 112	200	200

GANs; (2) attack success rate: indicates the accuracy of our membership inference in federated learning settings as we described in Section 4. In particular, the main task accuracy is the ratio of the correct classification of all samples through the global model.

5.2. Experimental Settings

5.2.1. Experimental Environment. We implemented the data augmentation and the membership inference in federated learning by using the PyTorch1.0, Tensorflow2.0, and Keras framework. All experiments are done on an RHEL7.5 server with NVidia Quadro P4000 GPU with 32 GB RAM and Ubuntu 16.04LTS OS. The Python version is 3.6. The router device is Redmi router AX6 supporting WIFI6. The experimental wireless network parameters are download rate≈32.81 Mbps, upload rate≈7.41 Mbps, delay≈10 ms, and signal strength≈-40 dBm. In the participant attack experiment, we set up five participants, one of whom is assumed as the adversary, while the remaining participants are benign. They are all subordinate to the same central server. In each round of the federated training, participants' local models are trained separately. Then, they synchronously upload their updates into a new global model. There are similar settings in the server attack experiment. One of the five participants is a victim, but their common server is malicious. The training process is still local training, and the victim will also participate in federated training, but its local model update is not synchronized with other participants, and this model is consistent with the affiliated server.

5.2.2. Model and Training Configurations. Considering the dataset used in our first part of the experiment about client-side inference, we applied a CNN-based model architecture to construct our membership inference classifier. Table 2 shows the neural network structure for two datasets. The model of MNIST consists of two convolutional layers and two dense layers. The kernel size of these convolution layers is 5×5. The number of filters for the first convolutional layer is 16 and for the second convolution layer is 32. The model for the CIFAR-10 dataset is set up as shown in Table 2. There are four convolutional layers with the 3×3 kernel size and 32×32 input shape. The number of filters for the first two convolutional layers is 32 and for the other convolution layers is 64. The activation function applied to all the neural network models is ReLU. In the second part of the experiment, the server-side inference, our GAN network structure refers to Table 3. We build a convolutional neural network consisting of three convolutional layers and three maximum pooling layers and set a fully connected layer at the end. The activation function is Tanh. The output of the model has 40 categories, corresponding to 40 categories of human faces. In the more complex situation of reconstructing face data, the generator G queries the discriminator D more times (the size of the adversary's training data divided by the batch size). Tanh in G is applied to set the output to the range of [-1, +1]. Since the AT&T image is large (64×64), G has an additional (5th) convolutional layer. G accepts a 100-dimensional uniform distribution as input and converts it to AT&T's image (64×64).

Table 2: Neural network structure.

Classifier	MNIST	CIFAR-10
Structure	Conv2D(16,5,5)+ReLU	Conv2D(32,3,3)+ReLU
	MaxPooling2D(2,2)	Conv2D(32,3,3)+ReLU
	Conv2D(32,5,5)+ReLU	MaxPooling2D(2,2)
	MaxPooling2D(2,2)	Conv2D(64,3,3)+ReLU
	FCL(1000)+ReLU	Conv2D(64,3,3)+ReLU
	FCL(10)+Softmax	MaxPooling2D(2,2)
		FCL(512)+ReLU
		FCL(10)+Softmax

The training configurations for two datasets are the following:

(i) Participants train MNIST dataset for epoch $E = 30$ with the initial learning rate $\eta = 0.01$

(ii) Participants train CIFAR-10 dataset for epoch $E = 60$ with the initial learning rate $\eta = 0.0001$

(iii) Participants train AT&T dataset for epoch $E = 60$ with the initial learning rate $\eta = 0.0002$

Besides, we run all the experiments for 400 communication rounds of the federated learning.

5.3. Performance of Data Augmentation. To illustrate the effectiveness of the data augmentation phase and data reconstruction using generative adversarial networks (GANs) in the malicious inference period, we visualize the process of sample generation. The total number of participants and the samples are not changed. The generator G is formatted as random noise with 100 lengths, and its output size is reshaped to 28×28 and 64×64 severally. In addition, we set the adversary to start generating samples after the global model accuracy reaches 80%.

Figures 7 and 8 show the visualization images of the sample reconstruction process as the number of iterations (communication rounds) increase and real samples for comparison. In Figure 7, the blurred contours of the reconstructed samples of 100 iterations can be recognized. As shown in the middle, in 400 iterations, the contours of the generator samples become clearer, because, with the update of the discriminator D, the performance of the generator G becomes better. Therefore, by deploying GANs, the adversary can successfully simulate real samples of all participants

Table 3: Neural network structure of GANs.

	Discriminator	Generator
Structure	Conv(1 ⟶ 32, 5 × 5)+Tanh MaxPooling(3 × 3,3,3) Conv(32 ⟶ 64, 5 × 5)+Tanh MaxPooling(2 × 2,2,2) Conv(64 ⟶ 128,5 × 5)+Tanh MaxPooling(2 × 2,2,2) Reshape(512) Linear(512 ⟶ 400) Tanh Linear(400 ⟶ 40) SoftMax	Conv(100 ⟶ 512,4 × 4) BatchNormalization ReLU Conv(512 ⟶ 256, × 4,2,2,1,1) BatchNormalization ReLU Conv(256 ⟶ 128, × 4,2,2,1,1) BatchNormalization ReLU Conv(128 ⟶ 64, × 4,2,2,1,1) BatchNormalization ReLU Conv(64 ⟶ 1, × 4,2,2,1,1) Tanh

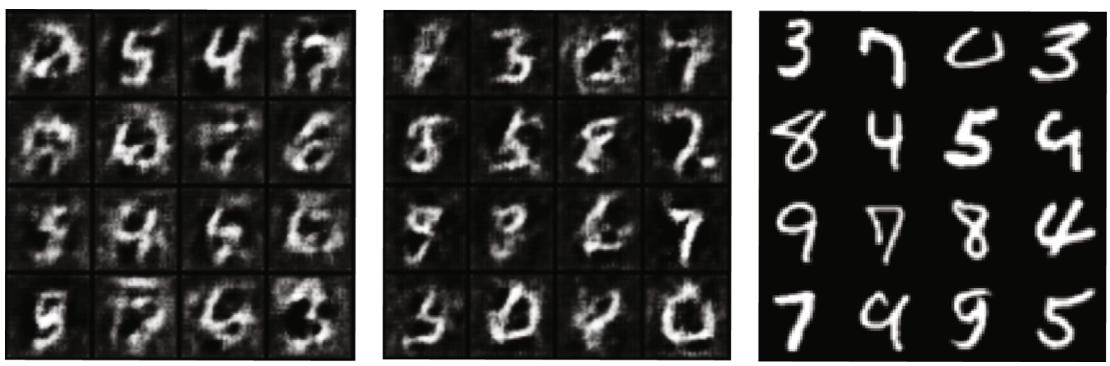


Figure 7: Reconstruction of MNIST based on GAN.

Figure 8: Reconstruction of AT&T based on GAN.

like the image on the right. Figure 8 selects the pictures of the three sequential stages generated by GANs in the malicious server using the isolation method. The four pictures in the first block preliminarily outline the cheek features of these people. The quality of pictures in the second block gradually becomes better. The four images in the third block are very close to the real samples on the far right.

5.4. Performance of Membership Inference. In the membership inference evaluation, the indexes are the main task (federated learning) and the accuracy of the membership inference.

As shown in Figure 9, with the secret membership inference attack, the accuracy of the models corresponding to three datasets, respectively, reaches 94.45%, 92.71%, and 84.48%, which are drawn with solid lines of three colors. In order to prove that the membership inference does not affect the normal training efficiency too much, we use the dotted lines with similar colors to draw the ordinary training process of these models as a baseline. Obviously, all three

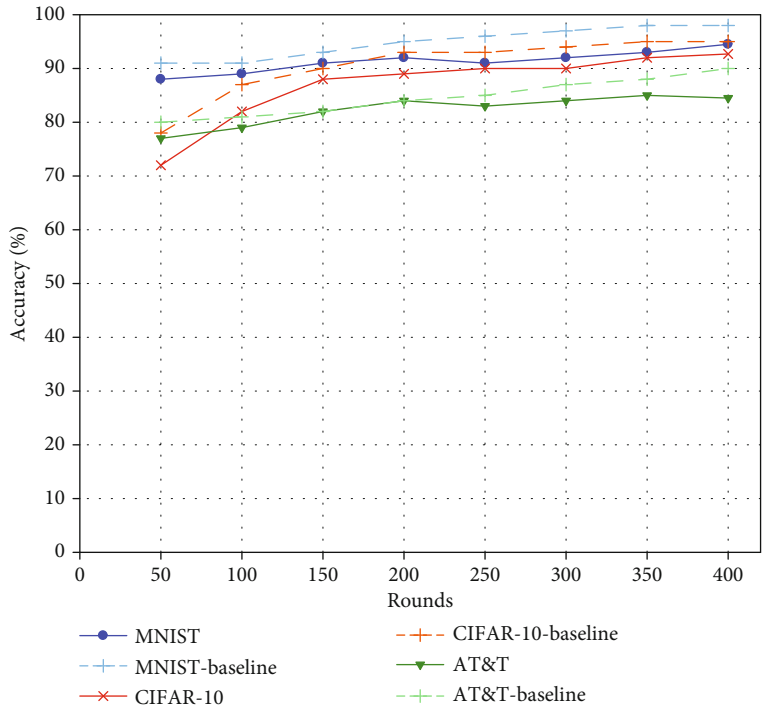


Figure 9: Effectiveness of main task.

models are accurate enough to complete the main task of correctly predicting all test data:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{11}$$

Simultaneously, behind the normal federated learning, the adversary is subtly acquiring enough fake samples through GANs and trained the attack model. After the membership inference, we evaluate the attack from the perspective of the class. Figure 10 illustrates our attack effectiveness on the three datasets. In order to fully demonstrate the inference results, we use the values of precision and recall to visualize the effect. The calculation formulas are shown in formulas (10) and (11), where TP means true positive, FN means false negative, and FP means false positive. We take the number of labels each participant has into account, supposing that the victim holds data with more than one label, which may disrupt the membership inference. We observe the effectiveness of attacks under the conditions of one label, two labels, three labels, and five labels. As is distinctly exhibited in the diagram, when participants hold more labels, the effect of membership inference launched by adversary pretending to be a participant will be worse, just as the values of precision and recall both drop from more than 80% to around 50%. But the adversary as a central server is not influenced by multiple classes, which benefits from restoring a designated participant's data pertinently. Contrast can be seen in these two modes, server-side's directive membership inference breaks through the bottleneck of being a client-side adversary that has no idea about the source of the fake data with multiple classes. We also draw some ROC curves to highlight this difference, where the variable is still the number of labels. Figures 11 and 12 show that when the target victim owns one or two labels around, the inference model on the client side can mark the member data as "IN" and the nonmember data as "OUT" with relative precision. However, with more classes, the false positive rate goes up a lot. Next, we will focus on trying to solve the problem of how to identify member data when there is data overlap between participants.

To highlight the advantages of our scheme, we compare the scheme with the active inference attacks based on the SGA method designed by Nasr et al. [8]. The inference accuracy of experiments based on the SGA method can reach about 76% on the CIFAR-100 dataset, which is close to the case where the participant holds one label in our CIFAR-10 experiment. But the biggest innovation is the attack objective. Nasr et al. [8] stated that the local adversary performs the inference against all other participants. In other words, this is the membership inference for the entire training data of the federated learning, which is not specific enough in our opinion. Our method can invoke membership inference directed to an individual participant under federated learning.

6. Discussion

This paper concentrates on the scenario of data instances where different participants do not have the data with the same class. This is reasonable because, in the federated

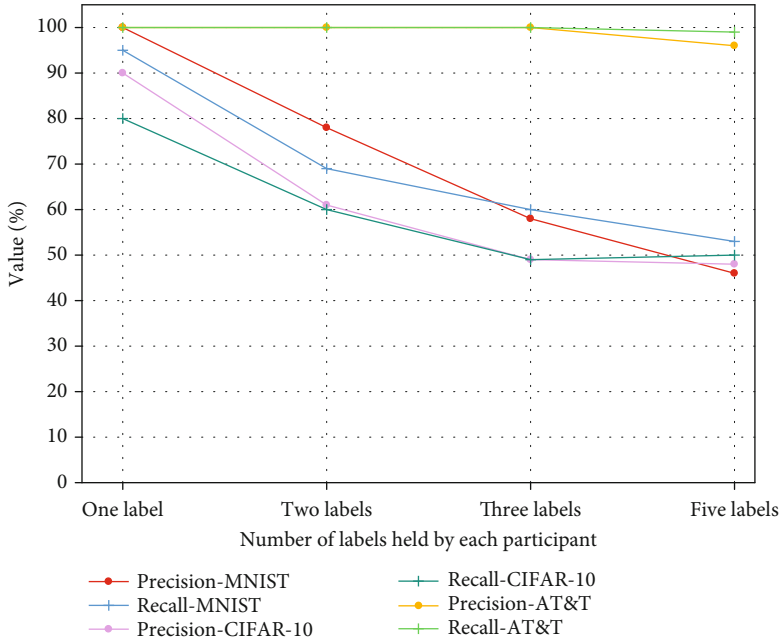


Figure 10: Effectiveness of inference task.

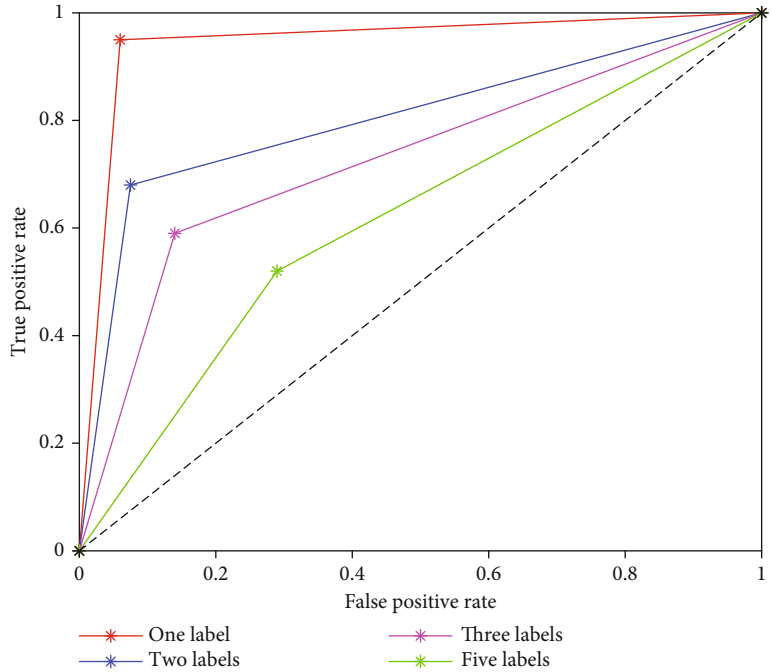


Figure 11: ROC curve of MNIST.

learning field, there are cases where different participants uphold a common training objective but possess very confidential (not shared) data with different labels. For example, if multiple variants of a virus are found in different countries and research institutions are reluctant to share data records with other foreign virologists for analysis, federated learning can play a huge role. At this time, the data with different labels are scattered across countries without any overlap, which is in line with our hypothetical scenario. However, what needs to be soberly aware is that the adversary's preparatory knowledge of the label information is not the end of our solution. Frankly speaking, in this paper, analyzing and classifying the ownership of target data are still at the primary stage of an ideal membership inference for federated learning. It is relatively easy for the adversary to get the answer. In fact, when some different data with the same label are scattered among the participants, the attack model can still give the data attribution, which is the advanced stage. Back to our proposed method, the ultimate goal of the data augmentation architecture we built is to simulate the

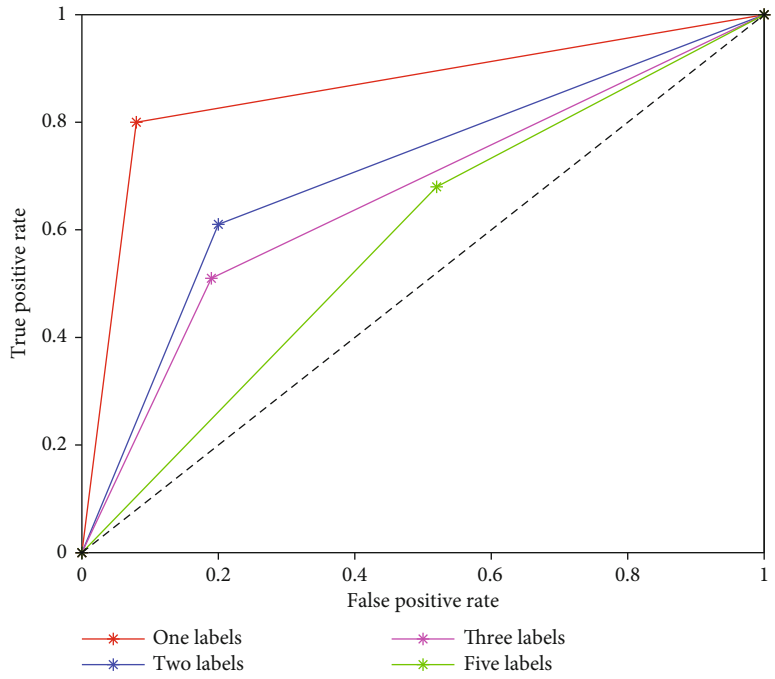


Figure 12: ROC curve of CIFAR-10.

victim's training data as much as possible to achieve advanced federated learning inference. At present, we have not been able to achieve this function. But we firmly believe that the introduced data augmentation architecture is the prototype of the possible technology for solving the advanced inference of federated learning in the future. We designed it as a reference to lay the foundation for future research. Currently, one potential breakthrough is to extract other "nontarget features" of the participants as the distinguishing elements [40], which need to be analyzed based on specific data. For a common example, the federated learning training bases on a globally distributed face dataset and the target representative are "whether wearing glasses or not." At this point, the face samples from the target area can be further filtered according to "complexion." Accordingly, the key feature of the membership inference model is changed to "complexion." Another possible way is the "conspiracy" that the server is colluding with the adversary or multiple adversaries are colluding. Demonstrating the feasibility of these conceptions will be put into our future work.

It is worth adding that, with the continuous training of massive data, the artificial intelligence algorithm model we adopted can better mine the intrinsic correlation of data, which may have better or even unexpected results.

Last but not least, in view of the instability and risk of the federated learning mechanism disclosed in this paper, we would like to propose some defense methods for the federated learning training process as our research contribution. We envision that the declaration information can be encrypted before participants start training. In this way, with the exception of the central server, the label information of all participants is unknown to each other, and it is difficult for the client-side adversary to distinguish the ownership of data. Obviously, asynchronous federated learning might be a more recommended privacy-preserving approach in more scenarios.

7. Conclusion

This paper is aimed at exploring an active and targeted membership inference attack model for federated learning in the wireless network environment. We proposed a fine-grained membership inference mode in the wireless environment, called the user-level membership inference. Given the traditional membership inference in centralized and distributed learning, we release the assumptions of some previous researches and launch membership inference from the client side and server side against a specific participant's data privacy. In order to counteract the influence of the privacy protection mechanism of federated learning posed to membership inference, where the adversary in the client side could only access an aggregated global model, we propose a data augmentation method using GANs with wireless monitor technique to obtain the high-quality generated samples with all labels. Not only that, we comprehensively study the case of a malicious server and successfully complete the membership inference task when participants hold data with multiple labels. Through the extensive experiments on three classic datasets, MNIST, CIFAR-10, and AT&T, we manage to prove that our proposed membership inference attack model can practically compromise the victim's privacy at the user level.

At last, we discuss the hypothetical premises of this paper and come up with some possible ideas. In future work, we will study these promising aspects, especially the

duplicated samples in the training sets, to prove their rationalities through experiments.

Data Availability

The data used in the experiment in this paper are all public datasets. They can be downloaded from the following link: MNIST: http://yann.lecun.com/exdb/mnist/, CIFAR-10: https://www.cs.toronto.edu/~kriz/cifar.html, and AT&T: https://www.kaggle.com/kasikrit/att-database-of-faces.

Disclosure

This paper is an extension of our previous work, which has been presented in the conference ICCCN2020 "Beyond Model-Level Membership Literacy: An Adversarial Approach in Federated Learning," DOI:10.1109/ICCCN49398.2020.9209744.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102000, in part by the National Natural Science Foundation of China under Grant 62172215, and in part by the Natural Science Foundation of Jiangsu Province (No. BK20200067).

References

[1] J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, and X. Zhu, "Deep learning towards mobile applications," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1385–1393, Vienna, Austria, July 2018.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[3] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018, http://arxiv.org/abs/1806.00582.

[4] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[5] S. L. Garfinkel, J. M. Abowd, and C. Martindale, "Understanding database reconstruction attacks on public data," *ACM Queue*, vol. 16, no. 5, pp. 28–53, 2018.

[6] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, San Jose, CA, USA, May 2017.

[7] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes, "Ml-leaks: model and data independent membership inference attacks and defenses on machine learning models," in *Network and Distributed Systems Security Symposium 2019. Internet Society*, San Diego, 2019.

[8] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning," in *2019 IEEE Symposium on Security and Privacy*, San Francisco.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds., , pp. 177–186, Physica-Verlag HD, 2010.

[10] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310–1321, New York, NY, USA, October 2015.

[11] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: efficient algorithms and tight error bounds," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473, Philadelphia, PA, USA, October 2014.

[12] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[13] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC '09: Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169–178, New York, NY, USA, May 2009.

[14] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: efficient homomorphic encryption for cross-silo federated learning," in *2020 USENIX Annual Technical Conference (USENIX ATC 20). USENIX Association*, pp. 493–506, July 2020, https://www.usenix.org/conference/atc20/presentation/zhang-chengliang.

[15] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: a client level perspective," 2017, http://arxiv.org/abs/1712.07557.

[16] S. Truex, N. Baracaldo, A. Anwar et al., "A hybrid approach to privacypreserving federated learning," in *AISec'19: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 1–11, New York, NY, USA, November 2019.

[17] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model," *Transactions on Data Privacy*, vol. 11, no. 1, pp. 61–79, 2018.

[18] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282, Oxford, UK, July 2018.

[19] Y. Long, V. Bindschaedler, L. Wang et al., "Understanding membership inferences on well-generalized learning models," 2018, http://arxiv.org/abs/1802.04889.

[20] J. Hayes, L. Melis, G. Danezis, and E. L. De Cristofaro, "Evaluating privacy leakage of generative models using generative adversarial networks," 2017, http://arxiv.org/abs/acs.CR/1705.07663.

[21] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 374–380, Rotorua, New Zealand, August 2019.

[22] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on*

Computer and Communications Security, pp. 603–618, New York, NY, USA, October 2017.

[23] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: userlevel privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520, Paris, France, April-May 2019.

[24] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proceedings Of The 25Th Usenix Security Symposium*, pp. 601–618, Austin, TX, USA, August 2016.

[25] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Inference attacks against collaborative learning," 2018, http://arxiv.org/abs/1805.04049.

[26] J. Konečný, M. M. HB, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," 2016, http://arxiv.org/abs/1610.05492.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.

[28] M. Martich, R. Hathaway, and K. Parsa, "Wireless access point," uS Patent App. 11/415,738, 2007.

[29] A. P. Jardosh, K. N. Ramach, K. C. Almeroth, and E. M. Belding-royer, "Understanding link-layer behavior in highly congested IEEE 802.11b wireless networks," in *E-WIND '05: Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, New York, NY, USA, August 2005.

[30] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *The Kluwer International Series in Engineering and Computer Science, vol 353*, T. Imielinski and H. F. Korth, Eds., pp. 153–181, Springer, Boston, MA, USA, 1996.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, http://arxiv.org/abs/1411.1784.

[32] M. Rabin, "Incorporating fairness into game theory and economics," *The American Economic Review*, vol. 83, no. 5, pp. 1281–1302, 1993.

[33] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[34] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: sequence generative adversarial nets with policy gradient," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, ser. AAAI'17*, pp. 2852–2858, San Francisco, 2017.

[35] G. Chaslot, E. Bakkes, I. Szita, and P. Spronck, "Montecarlo tree search: a new framework for game AI," *Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*, M. Mateas and C. Darken, Eds., , pp. 216-217, AAAI Press, Menlo Park, 2008.

[36] W. Fedus, I. J. Goodfellow, and A. M. Dai, "Maskgan: better text generation via filling in the," in *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April-May 2018https://openreview.net/forum?id=ByOExmWAb.

[37] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: revisited and enhanced," in *Applications and Techniques in Information Security*, L. Batten, D. S. Kim, X. Zhang, and G. Li, Eds., pp. 100–110, Springer Singapore, Singapore, 2017.

[38] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, 2012.

[39] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pp. 138–142, Sarasota, FL, USA, December 1994.

[40] G. Ateniese, G. Felici, L. V. Mancini, A. Spognardi, A. Villani, and D. Vitali, "Hacking smart machines with smarter ones: how to extract meaningful data from machine learning classifiers," 2013, http://arxiv.org/abs/1306.4447.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5570104, 13 pages
https://doi.org/10.1155/2021/5570104

Research Article

Incentives against Max-Min Fairness in a Centralized Resource System

Zheng Chen,[1] **Zhaoquan Gu**,[2] **and Yuexuan Wang**[3]

[1]*College of Computer Science and Technology, Zhejiang University, Hangzhou, China*
[2]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China*
[3]*College of Computer Science and Technology, Zhejiang University, Hangzhou, China*

Correspondence should be addressed to Zhaoquan Gu; zqgu@gzhu.edu.cn

Received 9 January 2021; Revised 2 June 2021; Accepted 11 September 2021; Published 18 October 2021

Academic Editor: Lin Wang

Resource allocating mechanisms draw much attention from various areas, and exploring the truthfulness of these mechanisms is a very hot topic. In this paper, we focus on the max-min fair allocation in a centralized resource system and explore whether the allocation is truthful when a node behaves strategically. The max-min fair allocation enables nodes receive appropriate resources, and we introduce an efficient algorithm to find out the allocation. To explore whether the allocation is truthful, we analyze how the allocation varies when a new node is added to the system, and we discuss whether the node can gain more resources if it misreports its resource demands. Surprisingly, if a node misrepresents itself by creating several fictitious nodes but keeps the sum of these nodes' resource demands the same, the node can achieve more resources evidently. We further present some illustrative examples to verify the results, and we show that a node can achieve 1.83 times resource if it misrepresents itself as two nodes. Finally, we discuss the influence of node's misrepresenting behavior in tree graph: some child nodes gain fewer resources even if their parent node gains more resources by creating two fictitious nodes.

1. Introduction

Fairly allocating limited resources (such as fossil fuels, clean water, and cyber source) has been widely studied. With so many competing needs and demands, lots of relevant works have been done. In early 1940s, considering the notion of proportionality, Steinhaus put forward a method to ensure n participants receive at least $1/n$ of each participant's own value for getting everything [1]. In a 1958 book of mathematical, Gamow and Stern [2] introduce the formal notion of "envy-freeness," in which participants preferring to keep their own allocation to swapping with other participants. Since then, research in this area has aroused widespread concern and has been applied to solve many important real-world problems [3–5].

Motivated by sharing economy concepts [6], nodes in a multiagent system may take strategic behaviors to gain larger utility. In [7], it proves that designing a truthful mechanism that guarantees a market equilibrium outcome in general is impossible when an agent has the incentive to misreport its information (such as its resource amount or demand). Some extant works also show that the truthfulness of resource allocating mechanisms cannot hold for a cheating node [8, 9]. However, the impossibility theorem proposed in [7] is not always correct for some restricted settings. For example, the proportional sharing mechanism is proved to be robust to agents' cheating, including an agent misreporting its resource or the connectivity with the rest of network [10, 11].

In this paper, we explore the truthfulness of resource allocation in a restricted setting. We consider a centralized resource system where only a central controller holds the resources that can be allocated to other nodes. Each node reports its resource demand to the controller who is in charge of allocating appropriate resources to the node. For example, each computer in a cluster of data center may require some resources while the central controller would make a fair resource allocation to all computers. In order to evaluate the fairness of a resource allocation, we adopt the concept of *max-min fairness* (a widely used concept in

networks, such as window flow central protocols [12], packer-switched network [13], and data network [14]) and explore whether the max-min fair allocation in the centralized resource system could be truthful if a node behaves strategically.

To begin with, we study the property of max-min fair allocation and introduce an efficient algorithm to find out the resource allocation mechanism. Afterwards, we explore how the nodes would be affected if a new node is connected to the controller. To study the truthfulness of the max-min fair allocation, we assume the new node could misreport its resource demand, and we analyze how this behavior changes the other nodes' allocation. In addition, we explore the effect of a novel splitting strategy that is proposed in recent work [15], where a node misrepresents itself by creating several fictitious agents but keeps the sum of the nodes' resource demand unchanged.

We summarize the contributions of the paper as follows:

(1) We introduce a simple and straightforward algorithm to find out the max-min fair allocation in a centralized resource system

(2) When a new node is connected to the controller, we analyze how much resource the new node could receive and how the other nodes' resource would be affected

(3) We explore the truthfulness of the max-min fair allocation. If the new node misreports its resource demand, we show that it cannot gain more resources through this strategic behavior

(4) We analyze the effect of node's misrepresenting behavior. By splitting itself into multiple nodes, a node could gain more resources in the allocation. Specifically, if a node is split into two nodes, it can achieve about two times resource compared to the allocation that it is honest under some special situations

(5) We analyze the effect of node's misrepresenting behavior for its child nodes on tree graph. Resources are passed down from the root node layer by layer. By splitting itself into two nodes, its child node gains fewer resources in the allocation under some special situations even if it received more resources from its parent node

We also present some numerical examples to illustrate these interesting results, and the examples show that the splitting strategy could indeed help a node gain about 1.83 times resource if it is split into two nodes (as shown in Table 1). What is more, some numerical examples are presented to show the influence of parent's splitting strategy on its child on tree graph. Some children gain fewer resources by its parent's splitting strategy.

The rest of the paper is organized as follows. The next section lists some related works about truthfulness of resource exchanging and allocation mechanisms. The preliminaries are introduced in Section 3. We present the efficient algorithm that finds out the max-min fair allocation in Section 4. When a new node is connected to the controller, we analyze the change of the allocation in Section 5. The truthfulness of the allocation is discussed in Section 6 if the new node misreports its resource, while the splitting strategy that could help gain more resources is analyzed in Section 7. The effect of node's splitting strategy for its child nodes on tree is analyzed in Section 8. Finally, we present some numerical examples in Section 9 and conclude the paper in Section 10.

Table 1: The max-min allocation after v_δ is split into two nodes.

Node i	m_i	t_i	x_{cv_i}	$x_{cv_i} \cdot t_i$	u_i
v_1	3	5	3	15	3
v_2	4	5	4	20	4
v_3	5	10	5	50	5
v_4	7	5	7	35	7
v_5	8	5	$85/12 \approx 7.083$	425/12	85/12
v_6	9	5	$85/12 \approx 7.083$	425/12	85/12
v_{p_1}	8	1	$85/12 \approx 7.083$	85/12	85/12
v_{p_2}	9	1	$85/12 \approx 7.083$	85/12	85/12

2. Related Work

The star network is generic and representative for many communication or economic networks. For example, the star network is commonly used in clustering of multityped heterogeneous networks [16], which extend the classical clustering on bityped heterogeneous networks [17]. Various networks have been mapped to the star networks [18], and we study the problem on the star network.

There are lots of researches about resource allocation in centralized system. One of the most critical resource allocating problems is the cake-cutting problem that is dividing a birthday cake among several children. During the World War II, Steinhaus, Knaster et al. [1] initiated the study of fair cake-cutting algorithms. How to help players achieve fair allocations is one of the most important strands in cake-cutting problem. Since then, a large number of related studies have emerged [5, 19, 20].

In resource exchange networks, max-min fairness is a simple and well-recognized approach to define a fair allocation in network [21–24], which has been widely adopted in economics [15, 25]. It is also used in ant colony algorithm for load balancing in could systems [26], and it is combined with convolutional neural networks for image classification [27].

In resource exchange networks, agents may be strategic, which makes the truthfulness of allocation mechanism being of paramount importance. In order to gain larger utility, an agent may misreport its true information, which may affect the market equilibrium. In [7], the authors prove that it is impossible to design a truthful mechanism that guarantees a market equilibrium outcome in general. In a Fisher market game with linear utility, agents can unilaterally deviate and

get a better payoff [8]. Even more specifically, agents may double its benefit by strategic behaviors [9]. Therefore, we also discuss about the truthfulness of the max-min fair allocation in a star network in this paper.

Generally, there are two types of agent strategic behaviors: misreporting its own resource amount and cheating on its connectivity with the rest of the network. By cheating on the connectivity, such as a missing edge or splitting a node into multiple nodes may affect the resource allocation mechanism. In [28], it examines the incentive problem in mechanism design in Internet applications or peer-to-peer systems when an agent misrepresents itself by creating several fictitious agents, which is also called false name bid. In [29], it measures how an agent gains if it deviates from its truthful behavior. In [30], it shows that agents' selfish behavior may contribute to the loss of social efficiency in Nash equilibrium in comparison to social optimality. In this paper, we adopt the splitting strategy to gain more resources in a star network.

3. Preliminaries

Consider a centralized resource system $G = (V, E)$ where a central node c holds the resources W_c that can be allocated to the other nodes $E \setminus \{c\} = \{v_1, v_2, \cdots, v_n\}$. For any node $v_i, i \in [1, n]$, denote m_i as the largest amount of resource v_i can receive (we regard it as resource demand or resource limitation). Considering an allocation mechanism, denote x_{cv_i} as the amount of resources node v_i receives from the central node, $x_{cv_i} \le m_i$; denote $t_i \ge 1$ as the resource transfer rate due to the link quality between the nodes. We assume that the resources would decay along the link, and thus, the central node actually allocates $x_{cv_i} \cdot t_i$ resources to node v_i (in traditional resource exchange or allocation mechanisms, we can set $t_i = 1$). Hence, $\sum_{i=1}^{n} x_{cv_i} \cdot t_i \le W_c$. Denote u_i as the utility of node v_i after receiving the resources from the central node, which can be represented by $u_i = f(x_{cv_i})$ where $f(\cdot)$ is a monotone increasing function. For simplicity, we assume $u_i = x_{cv_i}$ in this paper and denote $U = \{u_1, u_2, \cdots, u_n\}$ as the utility vector. To explore the effect of nodes' strategic behavior, we denote s_v as the reported strategy of node v, $\vec{s} = (s_{v_1}, s_{v_2}, \cdots, s_{v_n})$ as the nodes' reported strategy profile and $\vec{s}_{-v} = (s_{v_1}, s_{v_2}, \cdots, s_{v-1}, s_{v+1}, \cdots, s_{v_n})$ as the profile without node v. We denote $U_v(s_v, \vec{s}_{-v})$ as the utility of node v when it reports its strategy as s_v. The problem is to explore the truthfulness of the centralized max-min fair resource allocation problem. First, we are to find a max-min fair allocation vector $\overleftarrow{X}_{cv} = \{x_{cv_1}, x_{cv_2}, \cdots, x_{cv_n}\}$. To begin with, we introduce the property of max-min fairness.

Definition 1 (see [31]). A vector $\overleftarrow{x}$ is a max-min fair vector on set χ if and only if for all $\overleftarrow{y} \in \chi$, if there exists $s \in \{1, 2, \cdots, n\}$ such that $y_s > x_s$, there must exist $t \in \{1, 2, \cdots, n\}$ such that $y_t < x_t < x_s$.

The definition of a max-min fair vector reveals that increasing some element of a max-min fair vector $\vec{x}$ must be at the expense of decreasing some smaller (or equal) element of the vector.

Definition 2 (see [32]). A set χ is max-min achievable if there exists a max-min fair vector on χ.

The following proposition shows the uniqueness property of the max-min fair vector. If we can find out a max-min fair vector, it is the unique one.

Proposition 3 (see [31]). *If a max-min fair vector exists on a set χ, then it is unique.*

Definition 4. A mechanism is truthful if no agent can benefit strictly from misreporting its amount of resource irrespective of what is reported by other agents.

Formally, given a node v with true strategy s_v^* and the profile $\vec{s} = (s_v, \vec{s}_{-v})$, it holds that for any reported strategy s_v,

$$U_v\left(s_v, \vec{s}_{-v}\right) \le U_v\left(s_v^*, \vec{s}_{-v}\right). \tag{1}$$

In this paper, we first design a mechanism that finds out the max-min fair utility vector in a centralized resource system. Then, we explore the effect of two strategic behavior: misreporting and misrepresenting. We formalize the problems as:

Problem 5. Considering a centralized resource system $G = (V, E)$ where the central node holds resource W_c that can be allocated to n nodes $\{v_1, v_2, \cdots, v_n\}$. The problem is to design an allocation mechanism such that the utility vector $U = \{u_1, u_2, \cdots, u_n\}$ is the max-min fair vector.

The problem is equivalent to finding out the max-min fair allocation vector $\overrightarrow{X_{cv}} = \{x_{cv_1}, x_{cv_2}, \cdots, x_{cv_n}\}$. If one node v_i gets more resources through another allocation mechanism, another node v_j who receives fewer resources than v_i (i.e., $x_{cv_j} < x_{cv_i}$) would get resources fewer than x_{cv_j} in the new mechanism; this implies that increasing a node's resource of the max-min fair allocation would reduce a node's resource who originally receives fewer resources.

The max-min fair allocation mechanism could ensure the fairness among the nodes. When the system changes, such as a new node is connected to the central node, how many resources the new node should receive and will the other nodes be affected? This is the second problem we study. We formulate the problem as follows.

Problem 6. Considering a centralized resource system $G = (V, E)$ and the max-min fair allocation vector $\overrightarrow{X_{cv}^*} = \{x_{cv_1}^*, x_{cv_2}^*, \cdots, x_{cv_n}^*\}$, if a new node v_δ with resource limitation m_δ and transfer rate t_δ is connected to the central node, find out the max-min fair allocation vector $\overrightarrow{X_{cv}} = \{x_{cv_1}, x_{cv_2}, \cdots, x_{cv_n}, x_{cv_\delta}\}$ and how the nodes are affected.

In practice, some nodes may cheat to gain more resources. A simple cheating strategy is to misreport its resource limitation m_δ. We also analyze the truthfulness of the new node in Problem 6 and show that misreporting cannot help gain more resources (see Section 6). Therefore, we propose the splitting strategy to gain more resources as the following problem.

Problem 7. Considering a centralized resource system $G = (V, E)$, if a new node is connected to the central node with resource limitation m_δ and transfer rate t_δ, we denote the max-min fair allocation vector as $\overrightarrow{X_{cv}} = \{x_{cv_1}, x_{cv_2}, \cdots, x_{cv_n}, x_{cv_\delta}\}$. If the node is assumed to be split into two nodes v_{np_1}, v_{np_2} with resource limitations m_{np_1}, m_{np_2} ($m_{np_1} + m_{np_2} = m_\delta$) and the same transfer rate t_δ, find out the max-min fair allocation and analyze whether the node can gain more resources ($x_{cv_{np_1}} + x_{cv_{np_2}}$) compared to the received resource of the original allocation (x_{cv_δ}).

4. The Max-Min Fair Allocation Algorithm

We present the max-min fair allocation algorithm as Algorithm 1. The input of the algorithm is the resource limitation of each node $\vec{m} = \{m_1, m_2, \cdots, m_n\}$ and the resource transfer rates $\{t_1, t_2, \cdots, t_n\}$. The algorithm is to find the max-min fair vector $\vec{U} = \{u_1, u_2, \cdots, u_n\}$, which is equivalent to the allocation vector $\vec{X}_{cv} = \{x_{cv_1}, x_{cv_2}, \cdots, x_{cv_n}\}$.

As depicted in Algorithm1, $\vec{m}$ is firstly sorted by increasing order (line 3) and denotes the corresponding nodes as $\{v_1', v_2', \cdots, v_n'\}$ where $m_i' \le m_j'$ for any $i < j$. The resources are allocated to nodes from v_1' to v_n' according to lines 5-9. There are two situations: if the resources are adequate enough (line 5), node v_i' can receive the resources that are up to its limitation m_i' (line 6), or it is set as line 8. We show that the correctness of the algorithm by verifying the allocation vector $\vec{X}_{cv} = \{x_{cv_1}, \cdots, x_{cv_n}\}$ (corresponding to the utility vector $\vec{U} = \{u_1, u_2, \cdots, u_n\}$) is the max-min fair vector.

Theorem 8. *The output vector of Algorithm 1 is the max-min fair vector.*

Proof. For simplicity, we assume the input vector $\vec{m}$ is already ordered such that $v_i' = v_i$. We prove the theorem from two situations.□ □

For the first case, suppose $W_c \ge \sum_{j=1}^{n} t_j m_j$, which implies the central node has enough resources to ensure all nodes' resource demands can be satisfied. Thus, $\overrightarrow{X_{c_v}} = \{m_1, m_2, \cdots, m_n\}$. According to the definition of max-min vector, any element of the $\overrightarrow{X_{c_v}}$ can not be larger; hence, $\overrightarrow{X_{c_v}}$ is the max-min fair vector.

For the second case, suppose $W_c < \sum_{j=1}^{n} t_j m_j$; there must exists $i \in [1, n]$ such that $\forall j < i$, $x_{cv_j} = m_j$, and $x_{cv_i} = ((W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{j=i}^{n} t_j) < m_j$ since Algorithm 1 computes x_{cv_i} sequentially from $i = 1$ to $i = n$. Then, we derive the output of Algorithm 1 as

$$\vec{X}_{cv} = \left\{ m_1, \cdots, m_{i-1}, \frac{W_c - \sum_{j=1}^{i-1} t_j m_j}{\sum_{j=i}^{n} t_j}, \cdots, \frac{W_c - \sum_{j=1}^{i-1} t_j m_j}{\sum_{j=i}^{n} t_j} \right\}. \tag{2}$$

For any other allocation vector $\vec{X}_{cv}^{*} = \{x_{cv_1}^{*}, \cdots, x_{cv_n}^{*}\}$, if there exists $s \in [1, n]$ such that $x_{cv_s}^{*} > x_{cv_s}$, for any $t \in [1, n]$ satisfying $x_{cv_t} < x_{cv_s}$, suppose $x_{cv_t}^{*} \ge x_{cv_t}$; then, we have $x_{cv_1}^{*} t_1 + \cdots + x_{cv_n}^{*} t_n > x_{cv_1} t_1 + \cdots + x_{cv_n} t_n = W_c$, which leads to a contradiction since the allocation exceeds the amount of all resources W_c. Therefore, there exists $t \in [1, n]$ such that $x_{cv_t}^{*} < x_{cv_t} < x_{cv_s}$. By Definition 1, $\vec{X}_{cv}$ is the max-min fair vector.

Combining the two cases, the theorem holds.

From Proposition 3, the max-min fair vector is unique, and Algorithm 1 can find out the vector efficiently. We analyze the complexity of Algorithm 1 briefly. The complexity of sorting on line 3 is $O(n \log n)$, and the complexity of computing each x_{cv_i} in each step (lines 5-9) is $O(1)$. Thus, the complexity of the algorithm is $O(n \log n + n) = O(n \log n)$. In [32], the Max-Min Programming Algorithm (MP) is proposed to find the smallest element of the max-min fair vector by solving several linear programming. The complexity of MP algorithm is $O(n\mathrm{LP}(n, m))$ where $\mathrm{LP}(n, m)$ is the complexity of a linear program. Considering the problem we study in the paper, MP algorithm can be applied to find out the max-min allocation vector, but it works much slower than Algorithm 1. In the following parts, we adopt our algorithm to find out the max-min allocation mechanism when a new node is inserted or a node is split into multiple nodes.

5. Resource Allocation Varies for a New Node

In a centralized resource system $G = (V, E)$ with the max-min fair allocation vector $\overrightarrow{X_{cv}^{*}} = \{x_{cv_1}^{*}, x_{cv_2}^{*}, \cdots, x_{cv_n}^{*}\}$, if a new node v_δ is connected to the central node whose resource limitation is m_δ and transfer rate is t_δ, denote the max-min fair allocation vector of the new system as $\overrightarrow{X_{cv}} = \{x_{cv_1}, x_{cv_2}, \cdots, x_{cv_n}, x_{cv_\delta}\}$. We analyze the relationship between $\overrightarrow{X_{cv}^{*}}$ and $\overrightarrow{X_{cv}}$, which implies how the nodes' resources change regarding to the new inserted node.

Let R be the spare resources under the max-min fair allocation vector $\overrightarrow{X_{cv}^{*}}$ before the new node is inserted; we formulate it as

$$R = W_c - \sum_{j=1}^{n} x_{cv_j} t_j. \tag{3}$$

1: Input $\vec{m} = \{m_1, m_2, \cdots, m_n\}$, $\vec{t} = \{t_1, t_2, \cdots, t_n\}$
2: Output $\vec{U} = \{u_1, u_2, \cdots, u_n\}$
3: Sort $\vec{m}$ by increasing order as $\overrightarrow{m'} = \{m_1', m_2', \cdots, m_n',\}$ where $m_1', \leq m_j', \forall i < j$; denote the corresponding nodes as $\{v_1', v_2', \cdots, v_n'\}$;
4: **for** $i \longleftarrow 1$ to n **do**
5: **if** $\sum_{j=1}^{i-1} t_j' \cdot m_j' + m_i', \sum_{j=i}^{n} t_j' \leq W_c$ **then**
6: $x_{cv_i'} = m_i'$
7: **else**
8: **for** $k \longleftarrow i$ to n **do**
9: $x_{cv_k'} = W_c - \sum_{j=1}^{i-1} t_j' \cdot m_j' / \sum_{j=i}^{n} t_j'$, $u_k' = x_{cv_k'}$
10: **end for**
11: break;
12: **end if**
13: $u_i' = x_{cv_i'}$;
14: **end for**

Algorithm 1: The max-min fair allocation algorithm.

For simplicity, we suppose the nodes' resource limitations satisfy $m_1 \leq m_2 \leq \cdots \leq m_n$. We analyze how the max-min fair allocation changes when node v_δ is added to the system from two cases:

(i) *Case 1.* $R \geq t_\delta m_\delta$

(ii) *Case 2.* $0 \leq R < t_\delta m_\delta$

To begin with, we introduce a property of the max-min allocation mechanism which implies that the nodes with fewer received resources remain the same if a node with more resources does not change.

Property 9. In the max-min allocation, if there exists $k \in [1, n]$ such that $x_{cv_k} = m_k$, for all $i \in [1, k)$, $x_{cv_i} = m_i$.

Proof. Suppose there exists $i \in [1, k)$ satisfying $x_{cv_i} < m_i$, according to the allocation algorithm (Algorithm 1),

$$\sum_{j=1}^{i-1} t_j m_j + m_i \sum_{j=i}^{n} t_j > W_c, \tag{4}$$

since $i < k$, we can derive

$$\begin{aligned} \sum_{j=1}^{k-1} t_j m_j + m_k \sum_{j=k}^{n} t_j &= \sum_{j=1}^{i-1} t_j m_j + \sum_{j=i}^{k-1} t_j m_j + m_k \sum_{j=k}^{n} t_j \\ &\geq \sum_{j=1}^{i-1} t_j m_j + m_i \sum_{j=i}^{n} t_j > W_c, \end{aligned} \tag{5}$$

hence $x_{vu_k} < m_k$, which makes a contradiction.□ □

We analyze how the max-min fair allocation varies for the first case.

Theorem 10. *If $R \geq t_\delta m_\delta$, the new allocation holds $x_{cv_\delta} = m_\delta$, and $x_{cv_i} = m_i$ does not change when the new node v_δ is inserted.*

Proof. When $R \geq t_\delta m_\delta$, the spare resources are enough. It is obvious that $W_c \geq t_\delta m_\delta + \sum_{j=1}^{n} t_j m_j$. Suppose there exists i satisfying $m_{i-1} \leq m_\delta \leq m_k$ (if $m_\delta \leq m_1$, we let $m_0 = 0$, and such i exists), we derive

$$\sum_{j=1}^{i-1} t_j m_j + t_\delta m_\delta + m_\delta \sum_{j=i}^{n} t_j \leq t_\delta m_\delta + \sum_{j=1}^{n} t_j m_j \leq W_c. \tag{6}$$

According to lines 5-8 of Algorithm 1, $x_{cv_\delta} = m_\delta$. From Property 9, $x_{cv_j} = m_j$ when $1 \leq j \leq i-1$, $m_j \leq m_\delta$. For any $k \in [i, n]$ where $m_k \geq m_\delta$, we can compute

$$m_\delta t_\delta + \sum_{j=1}^{k-1} t_j m_j + m_k \sum_{j=k}^{n} t_j \leq m_\delta t_\delta + \sum_{j=1}^{n} t_j m_j \leq W_c. \tag{7}$$

Hence,

$$m_k \leq \frac{W_c - \sum_{j=1}^{k-1} t_j m_j - m_\delta t_\delta}{\sum_{j=k}^{n} t_j}, \tag{8}$$

from lines 5-8 of Algorithm 1, $x_{cv_k} = m_k$. The proof for the special cases $m_\delta \leq m_1$ and $m_\delta \geq m_n$ is similar, and we omit the details. Combining these aspects, the theorem holds.□ □

We then analyze how the max-min fair allocation varies for the second case when $0 \leq R < t_\delta m_\delta$. First, we find out a threshold r such that the nodes with resources fewer than r do not change their received resources (Lemma 11). Then, we analyze the change of nodes who received resources larger than r from two cases: $m_\delta > r$ (Lemma 12) and $m_\delta \leq$

r (Lemma 13). Finally, we compute the allocated resources of the new inserted node v_δ (Lemma14).

Lemma 11. *If $R < t_\delta m_\delta$, there exists $r = (W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{j=i}^{n} t_j + t_\delta$ such that any node $v_i, i \in [1, n]$ satisfying $m_i \le r$, $x_{cv_i} = m_i$ under the new max-min fair allocation vector.*

Proof. Suppose $m_1 \le m_2 \le \cdots \le m_{i-1} \le r \le m_i \le \cdots \le m_n$, we derive

$$\sum_{j=1}^{i-2} t_j m_j + m_{i-1} \sum_{j=i-1}^{n} t_j + t_\delta m_{i-1} \le W_c, \tag{9}$$

hence

$$m_{i-1} \le \frac{W_c - \sum_{j=1}^{i-2} t_j m_j}{t_\delta + \sum_{j=i-1}^{n} t_j}. \tag{10}$$

Then, we find the largest integer $i-1$ satisfying the above inequality, and we derive

$$r\left(\sum_{j=i}^{n} t_j + t_\delta\right) = W_c - \sum_{j=1}^{i-1} t_j m_j. \tag{11}$$

Thus, we figure out the threshold value r as

$$r = \frac{W_c - \sum_{j=1}^{i-1} t_j m_j}{\sum_{j=i}^{n} t_j + t_\delta}. \tag{12}$$

Since $x_{cv_{i-1}} = m_{i-1}$, for any node v_k with $m_k \le m_{i-1} \le r$, $x_{cv_k} = m_k$ by property 5. The lemma holds.□ □

It is easy to check that the node v_k with $m_k \le r$ received resources $x^*_{cv_k} = m_k$ in the original allocation vector before the new node is inserted. Finding out the threshold value r, these nodes' received resources do not change. For the nodes v_k with $m_k < r$, we analyze them from two cases: $m_\delta > r$ and $m_\delta \le r$.

Lemma 12. *If $R < t_\delta m_\delta$, $m_\delta > r = (W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{j=i}^{n} t_j + t_\delta$, for any node $v_i, i \in [1, n]$ satisfying $m_i > r$, $x_{cv_i} < x^*_{cv_i} \le m_i$ under the new max-min fair allocation vector.*

Proof. First, we show that $x_{cv_i} < m_i$ under the new max-min fair allocation for any node v_i satisfying $m_i > r$. This means the threshold r divides the nodes into two subsets; one subset contains nodes that receive resources up to its limitation as Lemma 11, while the other subset consists of nodes with resource limitation larger than r and cannot receive resources up to its limitation.

Suppose there exists a node v_k such that $m_k > r$ and $x_{cv_k} = m_k$. According to Property 9, node v_i ($v \in [1, k-1]$) receives resources $x_{cv_i} = m_i$.

There are two cases: $m_k \ge m_\delta$ and $m_k < m_\delta$. For the first case that $m_k \ge m_\delta$, we drive

$$\sum_{j=1}^{k-1} t_j m_j + m_\delta t_\delta + m_k \sum_{j=k}^{n} t_j > \sum_{j=1}^{k-1} t_j m_j + r t_\delta + r \sum_{j=k}^{n} t_j = W_c, \tag{13}$$

which is a contradiction. When $m_\delta > m_k$, we can deduce a contradiction similarly. Therefore, $x_{cv_i} < m_i$ for node v_i with $m_i > r$.

Then, we show that node v_i with $m_i > r$ receives fewer resources compared to the original max-min fair allocation; that is, $x_{cv_i} < x^*_{cv_i}$. Denote $x^*_{cv_i} = \alpha$, suppose $x_{cv_i} \ge x^*_{cv_i} = \alpha$, we can derive

$$\sum_{j=1}^{i-1} t_j m_j + m_\alpha t_\delta + \alpha \sum_{j=i}^{n} t_j > \sum_{j=1}^{i-1} t_j m_j + \alpha \sum_{j=i}^{n} t_j > W_c, \tag{14}$$

which is a contradiction.

Combining the two sides, $x_{cv_i} < x^*_{cv_i} \le m_i$ holds.□ □

Lemma 12 implies that if the new node's resource limitation is large $m_\delta > r$, all nodes with large resources limitation ($m_i > r$) have to reduce its received resources. If $m_\delta \le r$, fewer nodes have to reduce its resources, and we show it in the next lemma.

Lemma 13. *If $R < t_\delta m_\delta$, $m_\delta \le r = ((W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{j=i}^{n} t_j + t_\delta)$, there exists $r_1 \ge r$ such that the following two results hold:*

(i) If node v_i satisfies $m_i > r_1$, $x_{cv_i} < x^*_{cv_i} \le m_i$

(ii) If node v_i satisfies $m_i \le r_1$, $x_{cv_i} = x^*_{cv_i} = m_i$

Proof. If $m_\delta \le r$, there exists $i_1 \le i_2$ such that $m_1 \le m_2 \le \cdots \le m_{i_1-1} \le m_\delta \le \cdots \le m_{i_2-1} \le r \le m_{i_2} \le \cdots \le m_n$; we have

$$\sum_{j=1}^{i_2-1} t_j m_j + m_\delta t_\delta + m_{i_2-1} \sum_{j=i_2}^{n} t_j \le \sum_{j=1}^{i_2-1} t_j m_j + r\left(t_\delta + \sum_{j=i_2}^{n} t_j\right) \le W_c, \tag{15}$$

thus $x_{cv_{i_2-1}} = m_{i_2-1}$. According to Property 9, $x_{cv_k} = m_k$ if $k \le i_2 - 1$, which implies $x_{cv_\delta} = m_\delta$. Since $R < t_\delta m_\delta$, nodes with more resources would reduce its resources, and there must exist $r_1 \ge m_{i_2-1}$ such that $x_{cv_k} = x^*_{cv_k} = m_k$ if $m_k \le r_1$ (nodes do not change their resources), and $x_{cv_k} < x^*_{cv_k} \le m_k$ if $m_k > r_1$ (nodes reduce their resources).

Next, we show that the value $r_1 \ge r$. It is clear that if $r_1 \ge m_{i_2}$, $r_1 > r$. Suppose $m_{i_2-1} \le r_1 < m_{i_2}$, we derive

$$r_1 \sum_{j=i_2}^{n} t_j + m_\delta t_\delta = W_c - \sum_{j=1}^{i_2-1} t_j m_j, \tag{16}$$

combining the formulation of r, we show that

$$(r_1 - r)\sum_{j=i_2}^{n} t_j = (r - m_\delta)t_\delta \geq 0. \quad (17)$$

Thus, $r_1 \geq r$ holds. It is easy to check node v_k with $m_k \leq r_1$ receives resources $x^*_{cv_k} = m_k$ in the original allocation. Finding out the threshold value r_1, these nodes' received resources do not change. Under the new max-min fair allocation, the node v_k with $m_k > r_1$ receives resources $x_{cv_k} < x^*_{cv_k} \leq m_k$, and the proof is similar to that of Lemma 12.□ □

Lemma 14. *If $R < t_\delta m_\delta$, there exists $r = (W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{j=i}^{n} t_j + t_\delta$ such that $x_{cv_\delta} = m_\delta$ if $m_\delta \leq r$, and $x_{cv_\delta} < m_\delta$ if $m_\delta > r$.*

Proof. According to the proof of Lemma 13, $x_{cv_\delta} = m_\delta$ if $m_\delta \leq r$. We show that $x_{cv_\delta} < m_\delta$ if $m_\delta > r$.

Suppose $m_1 \leq m_2 \leq \cdots \leq m_{i_1-1} \leq r \leq \cdots \leq m_{i_2-1} \leq m_\delta \leq m_{i_2} \leq \cdots \leq m_n$, if $x_{cv_\delta} = m_\delta$, we derive

$$\sum_{j=1}^{i_2-1} t_j m_j + m_\delta t_\delta + m_\delta \sum_{j=i_2}^{n} t_j > \sum_{j=1}^{i_2-1} t_j m_j + r t_\delta + r \sum_{j=i}^{n} t_j = W_c, \quad (18)$$

which is a contradiction. Thus, the lemma holds.□ □

Combining Lemmas 11, 12, 13, and 14, we conclude the situation when the spare resources are inadequate $0 \leq R < t_\delta m_\delta$ as Theorem 15.

Theorem 15. *If $0 \leq R < t_\delta m_\delta$, there exist two thresholds r, r_1 as Lemma 11 and Lemma 13. If $m_\delta > r$, the nodes satisfy:*

$$\begin{cases} x_{cv_i} = x^*_{cv_i} & \text{if } m_i \leq r, \\ x_{cv_i} < x^*_{cv_i} \leq m_i & \text{if } m_i > r, \\ x_{cv_\delta} < m_\delta. \end{cases} \quad (19)$$

If $m_\delta \leq r$, the nodes satisfy:

$$\begin{cases} x_{cv_i} = x^*_{cv_i} & \text{if } m_i \leq r_1, \\ x_{cv_i} < x^*_{cv_i} \leq m_i & \text{if } m_i > r_1, \\ x_{cv_\delta} = m_\delta. \end{cases} \quad (20)$$

6. Truthfulness against the Misreporting Strategy

In this section, we analyze the truthfulness of the max-min fair allocation. Denote $\vec{X}_{cv} = (x_{cv_1}, x_{cv_2}, \cdots, x_{cv_{i-1}}, x_{cv_\delta}, x_{cv_i}, x_{cv_n})$ as the allocation after the new node v_δ is inserted. Denote $\vec{M} = \{m_1, m_2, \cdots, m_{i-1}, m_\delta, m_i, \cdots, m_n\}$ as the resource limitation vector where the new node's limitation suits $m_{i-1} \leq m_\delta \leq m_i$. Denote $\vec{M}_{-u} = \{m_1, m_2, \cdots, m_n\}$ as the vector before the new node is inserted. We study how the allocation varies when the new node misreports its resource limitation.

As shown in the previous section, the allocated resources x_{cv_δ} of the new node v_δ can be regarded as a function of m_δ when $\vec{M}_{-u}$ is fixed. Suppose the new node misreports its resource limitation as m_{nt}; we analyze the effect from two situations $R \geq t_\delta m_\delta$ (the spare resources are adequate) or $R < t_\delta m_\delta$.

Lemma 16. *When $R \geq t_\delta m_\delta$, if the new node reports fewer resources ($m_{nt} \in (0, m_\delta]$), $x_{cv_\delta} = m_{nt}$; if the node reports more resources ($m_{nt} \in (m_\delta, +\infty]$), at most $r - m_\delta$ resources would be wasted where $r = (W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{i}^{n} t_j + t_\delta$.*

Proof. According to Theorem 2, $x_{cv_\delta} = m_{nt}$ for any $m_{nt} \in (0, m_\delta]$ under the max-min fair allocation; this implies that the node would receive fewer resources if it reports fewer. If the node reports more resources, the central node may allocate more resources, but the resources beyond its true limitation would be wasted. From Lemma 11, the largest amount of resources that the new node could receive is $r = (W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{i}^{n} t_j + t_\delta$; therefore, at most $r - m_\delta$ resources would be wasted.□ □

Lemma 17. *When $R < t_\delta m_\delta$, there exists a real number r_2 satisfying the following conditions:*

(i) If $m_\delta \leq r_2$, for any $m_{nt} \in [0, m_\delta]$, x_{cv_δ} is a monotonically increasing function of m_{nt} and no resource is wasted; for any $m_{nt} \in (m_\delta, +\infty)$, at most $r_2 - m_\delta$ resources are wasted

(ii) If $m_\delta > r_2$, for any $m_{nt} \in [0, m_\delta]$, $x_{cv_\delta} = m_{nt}$; for any $m_{nt} \in [r_2, +\infty)$, $x_{cv_\delta} = r_2 < m_\delta$; therefore, no resources are wasted

Proof. To begin with, we derive the value of r_2 by the following three cases:

(1) If $R \geq t_\delta m_n$, $r_2 = R/t_\delta$

(2) If $m_1(\sum_{j=1}^{n} t_j + t_\delta) \geq W_c$, $r_2 = W_c/\sum_{j=1}^{n} t_j + t_\delta$

(3) If $R < t_\delta m_n$ and $m_1(\sum_{j=1}^{n} t_j + t_\delta) < W_c$, from Lemma 11 to derive $r_2 = (W_c - \sum_{j=1}^{i-1} t_j m_j)/\sum_{j=i}^{n} t_j + t_\delta$

□ □

For simplicity, we only show the proof sketch of the third case. According to Lemma 11, when $m_{nt} \in [0, r_2]$, $x_{cv_\delta} = m_{nt}$, thus x_{cv_δ} is a monotonically increasing function of m_{nt}. Regardless of what value m_{nt} the new node reports, the received resources cannot be larger than r_2. Hence, if $m_\delta \geq r_2$ and $m_{nt} \in [r_2, +\infty)$, $x_{cv_\delta} = r_2 < m_\delta$, and no resource is wasted. However, when $m_\delta \leq r_2$ and $m_{nt} \in [r_2, +\infty)$, the new node receives more resources than its true limitation, causing at most $r_2 - m_\delta$

resources wasted. Thus, the lemma holds. The proof of the other two cases are similar, and we omit the details.

According to Lemmas 16 and 17, the new node misreport that its resource limitation cannot gain more resources, and some resources would be wasted under some circumstances.

7. Truthfulness against the Spitting Strategy

The previous section shows that a node cannot gain more resources by misreporting its resource limitation. In this section, we propose a novel method which may help the node receive more resources. Suppose a new node's resource limitation is m_δ; it reports to the central node that two nodes are inserted, and the sum of their resource limitations is m_δ. This implies that the node is assumed to be split into two nodes with resource limitations m_{np_1}, m_{np_2}, and $m_{np_1} + m_{np_2} = m_\delta$; we call this *splitting strategy*. We analyze when the node could achieve more resources through this method.

There are four cases according to the values of m_δ (the resource limitation of the new node), m_n (the largest resource limitation of the original nodes), and R (the spare resources before the new node is inserted, Equation (3)):

(i) Case 1: $R \geq t_\delta m_\delta$

(ii) Case 2: $R < t_\delta m_\delta$ and $R \geq 2t_\delta m_n$

(iii) Case 3: $R < t_\delta m_\delta$ and $R < 2t_\delta m_n$

In this section, we show that the new node cannot gain more resources for the first two cases (Lemmas 18 and 19), but it can gain more resources for cases 3 (Lemma 20). Denote the two virtual nodes of v_δ as v_{np_1}, v_{np_2} and the received resources as $x_{cv_{np_1}}, x_{cv_{np_2}}$, respectively. Denote the received resources of node v_δ as x_{cv_δ} if it does not report two nodes. We show the following lemmas:

Lemma 18. *If $R \geq t_\delta m_\delta$, the splitting strategy has no effect to the new node, i.e., $x_{cv_{np_1}} + x_{cv_{np_2}} = x_{cv_\delta}$.*

Proof. Without loss of generality, suppose $m_{np_1} \leq m_{np_2}$ and $m_1 \leq m_2 \leq \cdots \leq m_{k_1} \leq m_{np_1} \leq \cdots \leq m_{np_2} \leq m_{k_2} \leq \cdots \leq m_n$. When node v_{np_1} is inserted, we derive

$$\sum_{j=1}^{k_1} t_j m_j + m_{np_1} t_\delta + \sum_{k_1+1}^{n} t_j m_j < \sum_{j=1}^{n} t_j m_j + m_\delta t_\delta \leq W_c, \quad (21)$$

then $x_{cv_{np_1}} = m_{np_1}$. When node v_{np_2} is inserted, we derive

$$\begin{aligned} &\sum_{j=1}^{k_1} t_j m_j + m_{np_1} t_\delta + \sum_{j=k_1+1}^{k_2-1} t_j m_j + m_{np_2} t_\delta + \sum_{j=k_2}^{n} t_j m_j \\ &< \sum_{j=1}^{n} t_j m_j + \left(m_{np_1} + m_{np_2}\right) t_\delta \leq \sum_{j=1}^{n} t_j m_j + m_\delta t_\delta \leq W_c, \end{aligned} \quad (22)$$

then $x_{cv_{np_2}} = m_{np_2}$. Therefore, $x_{cv_{np_1}} + x_{cv_{np_2}} = m_{np_1} + m_{np_2} = m_\delta = x_{cv_\delta}$, which implies that the new node cannot gain more resources through the splitting strategy.□ □

Lemma 19. *If $R < t_\delta m_\delta$ and $R \geq 2t_\delta m_n$, the splitting strategy has no effect to the new node, i.e., $x_{cv_{np_1}} + x_{cv_{np_2}} = x_{cv_\delta}$; thus, there is still no need to split the node.*

Proof. Since $R < t_\delta m_\delta$ and $R \geq 2t_\delta m_n$, we derive $t_\delta m_\delta > 2t_\delta m_n$, which implies $m_\delta > 2m_n$. Without loss of generality, suppose $m_{np_1} \leq m_{np_2}$, there are two cases

(1) $m_{np_1} \leq m_n < m_{np_2}$ and $m_{np_1} + m_{np_2} = m_\delta > 2m_n$

(2) $m_n < m_{np_1} \leq m_{np_2}$ and $m_{np_1} + m_{np_2} = m_\delta > 2m_n$

If $m_{np_1} \leq m_n < m_{np_2}$, suppose $m_1 \leq m_2 \leq \cdots \leq m_{i-1} \leq m_{np_1} \leq m_i \leq \cdots \leq m_n < m_{np_2}$; notice that

$$W_c \geq \sum_{j=1}^{n} t_j m_j + 2t_\delta m_n > \sum_{j=1}^{i-1} t_j m_j + t_\delta m_{np_1} + m_{np_1} \sum_{j=i-1}^{n} t_j. \quad (23)$$

When node v_{np_1} is inserted, $x_{cv_{np_1}} = m_{np_1}$. When node v_{np_2} is inserted, from $t_\delta m_\delta > W_c - \sum_{j=1}^{n} m_j t_j$, we have

$$t_\delta m_{np_2} > W_c - t_\delta m_{np_1} - \sum_{j=1}^{n} t_j m_j, \quad (24)$$

thus the allocated resources are

$$x_{cv_{np_2}} = \frac{W_c - t_\delta m_{np_1} - \sum_{j=1}^{n} t_j m_j}{t_\delta}. \quad (25)$$

Combining two nodes' resources, $x_{cv_{np_1}} + x_{cv_{np_2}} = m_{np_1} + ((W_c - t_\delta m_{np_1} - \sum_{j=1}^{n} t_j m_j)/t_\delta) = ((W_c - \sum_{j=1}^{n} t_j m_j)/t_\delta) = x_{cv_\delta}$.

For the second case, $m_n < m_{np_1} \leq m_{np_2}$ and $m_{np_1} + m_{np_2} > 2m_n$; notice that after the two nodes are inserted, for any node $v_i, i \in [1, n]$, $x_{cv_i} = m_i$. If $W_c - \sum_{j=1}^{n} t_j m_j \geq 2t_\delta m_{np_1}$, we derive

$$\begin{cases} x_{cv_{np_1}} = m_{np_1}, \\ x_{cv_{np_2}} = \dfrac{W_c - \sum_{j=1}^{n} t_j m_j - t_\delta m_{np_1}}{t_\delta}. \end{cases} \quad (26)$$

Therefore, $x_{cv_{np_1}} + x_{cv_{np2}} = ((W_c - \sum_{j=1}^{n} t_j m_j)/t_\delta) = x_{cv_\delta}$. If $W_c - \sum_{j=1}^{n} t_j m_j \leq 2t_\delta m_{np_1}$, we derive $x_{cv_{np_1}} = x_{cv_{np_2}} = (W_c - \sum_{j=1}^{n} t_j m_j)/2t_\delta$, thus $x_{cv_{np_1}} + x_{cv_{np_2}} = ((W_c - \sum_{j=1}^{n} t_j m_j)/t_\delta) = x_{cv_\delta}$. Combining these cases, the lemma holds.□ □

Lemma 20. *If $R < t_\delta m_\delta$ and $R < 2t_\delta m_n$, the new node can increase the received resources through the splitting method.*

For some special situations, the received resources could be about two times, i.e., $x_{cv_{np_1}} + x_{cv_{np_2}} \approx 2x_{cv_\delta}$.

Proof. If $R < t_\delta m_\delta$ and $R < 2t_\delta m_n$, there exists m_* such that the largest amount of resources two nodes can receive is 2 m_*. Suppose $m_1 \le m_2 \le \cdots \le m_{i-1} \le m_* \le m_i \le \cdots \le m_n$; similar as Lemma 11, we derive

$$\sum_{j=1}^{i-2} t_j m_j + m_{i-1} \sum_{j=i-1}^{n} t_j + 2t_\delta m_{i-1} \le W_c. \tag{27}$$

Thus, $m_{i-1} \le (W_c - \sum_{j=1}^{i-2} t_j m_j)/(2t_\delta + \sum_{j=i-1}^{n} t_j)$, and we can find the largest value $i-1$ such that $m_{i-1} \le m_*$ and $m_i \ge m_*$. Since $\sum_{j=1}^{i-1} t_j m_j + m_* \sum_{j=i}^{n} t_j + 2t_\delta m_* = W_c$, we derive

$$m_* = \frac{W_c - \sum_{j=1}^{i-1} t_j m_j}{2t_\delta + \sum_{j=i}^{n} t_j}. \tag{28}$$

According to Algorithm 1, if $m_* \le m_{np_1} \le m_{np_2}$, $x_{cv_{np_1}} = x_{cv_{np_2}} = m_*$. If $m_{np_1} \le m_* \le m_{np_2}$, we show that the sum of two nodes' resources is no more than $2m_*$.

Suppose $m_1 \le m_2 \le \cdots \le m_{i_1} \le m_{np_1} \le \cdots \le m_* \le \cdots \le m_{i_2} \le m_{np_2} \le \cdots \le m_n$, it is easy to get $x_{cv_{np_1}} = m_{np_1}$ and $x_{cv_{np_2}} \ge m_*$. We analyze the maximum value $x_{cv_{np_2}}$ can achieve. Combining the inequality $\sum_{j=1}^{i_1} t_j m_j + t_\delta m_{np_1} + \sum_{j=i_1+1}^{i_2} t_j m_j + x_{cv_{np_2}} t_\delta + x_{cv_{np_2}} \sum_{j=i_2+1}^{n} t_j \le W_c$. and the equation $\sum_{j=1}^{i-1} t_j m_j + m_* \sum_{j=i}^{n} t_j + 2t_\delta m_* = W_c$, we derive

$$\begin{aligned}\left(m_{np_1} + x_{cv_{np_2}} - 2m_*\right) t_\delta &\le \left(\sum_{j=1}^{i-1} t_j m_j + m_* \sum_{j=i}^{n} t_j\right) \\ &\quad - \left(\sum_{j=1}^{i_2} t_j m_j + x_{cv_{np_2}} \sum_{j=i_2+1}^{n} t_j\right) \\ &\le 0.\end{aligned} \tag{29}$$

Hence, $x_{cv_{np_1}} + x_{cv_{np_2}} \le m_{np_1} + x_{cv_{np_2}} \le 2m_*$. We then analyze when the $x_{cv_{np_1}} + x_{cv_{np_2}} = 2m_*$.

If $m_{np_2} > m_*$, $i \le i_2$ and $(m_{np_1} + x_{cv_{np_2}} - 2m_*)t_\delta < 0$ (Equation (29)), the sum of two nodes' resources cannot be $2m_*$. When $x_{cv_{np_2}} = m_*$, we can derive $x_{cv_{np_1}} + x_{cv_{np_2}} = m_{np_1} + x_{cv_{np_2}} = 2m_*$. If the node is not split, the received resources is $x_{cv_\delta} = (W_c - \sum_{j=1}^{i_1-1} t_j)/(t_\delta + \sum_{j=i_1}^{n} t_j)$. Since $m_* = (W_c - \sum_{j=1}^{i-1} t_j m_j)/(2t_\delta + \sum_{j=i}^{n} t_j)$ (see Equation (28)), if $t_\delta << \sum_{j=1}^{n} t_j$, we derive $i_1 = i$, and the received resources of two split nodes are $2m_* \approx 2x_{cv_\delta}$, which implies that the node can receive about 2 times resources compared to the original allocation. Therefore, the lemma holds.□ □

From the lemma, we know that the new node can achieve more resources by the splitting strategy. For some special situations, the node can gain appropriate two times resources compared to a honest node. Combining Lemmas 18, 19, and 20, we conclude the theorem as follows:

If the spare resources are adequate ($R \ge t_\delta m_\delta$) or inadequate ($R < t_\delta m_\delta$) but suits $R \ge 2t_\delta m_n$, the splitting strategy has no effect on the new node. If the spare resources suit $R < t_\delta m_\delta$ and $R < 2t_\delta m_n$, the node can gain more resources ($x_{cv_{np_1}} + x_{cv_{np_2}} > x_{cv_\delta}$), and in some special situations, the resources can be about two times ($x_{cv_{np_1}} + x_{cv_{np_2}} \approx 2x_{cv_\delta}$).

8. Truthfulness against Splitting Strategy in Tree Graph

In the previous section, we show that a node can gain more resources by the splitting strategy in the centralized resource system. However, in the hierarchy, resources are passed down layer by layer. The node cannot obtain resources from the original node that owns the resources but receives resources from other adjacent nodes. In this section, we will study how the splitting behavior of the parent node affects the child nodes. Suppose a node v_r has several child nodes $v_{p_i}, i \in [1, k_1]$, the limitation of which is m_{p_i}. Node v_{p_t} has several child node $v_{c_j}, j \in [1, k_2]$. Node v_{p_θ} reports to its parent node v_r that two nodes $v_{p_\theta}^1$ and $v_{p_\theta}^2$ are inserted and the sum of their resource limitation is m_{p_t}. We analyze the influence of node v_{p_i}'s splitting strategy on its child node v_{c_j}'s received resources. x_{ij} is the resources node v_i allocates to node v_j. W_r is the resources of node v_r. Let R_r be the spare resources after node v_r allocates its resources to its child node v_{c_i} except node v_{c_i}. We formulate it as $R_r = W_r - \sum_{i \in [1, p_\theta - 1] \bigcup [p_\theta + 1, k_1]} x_{p_i} t_{p_i}$.

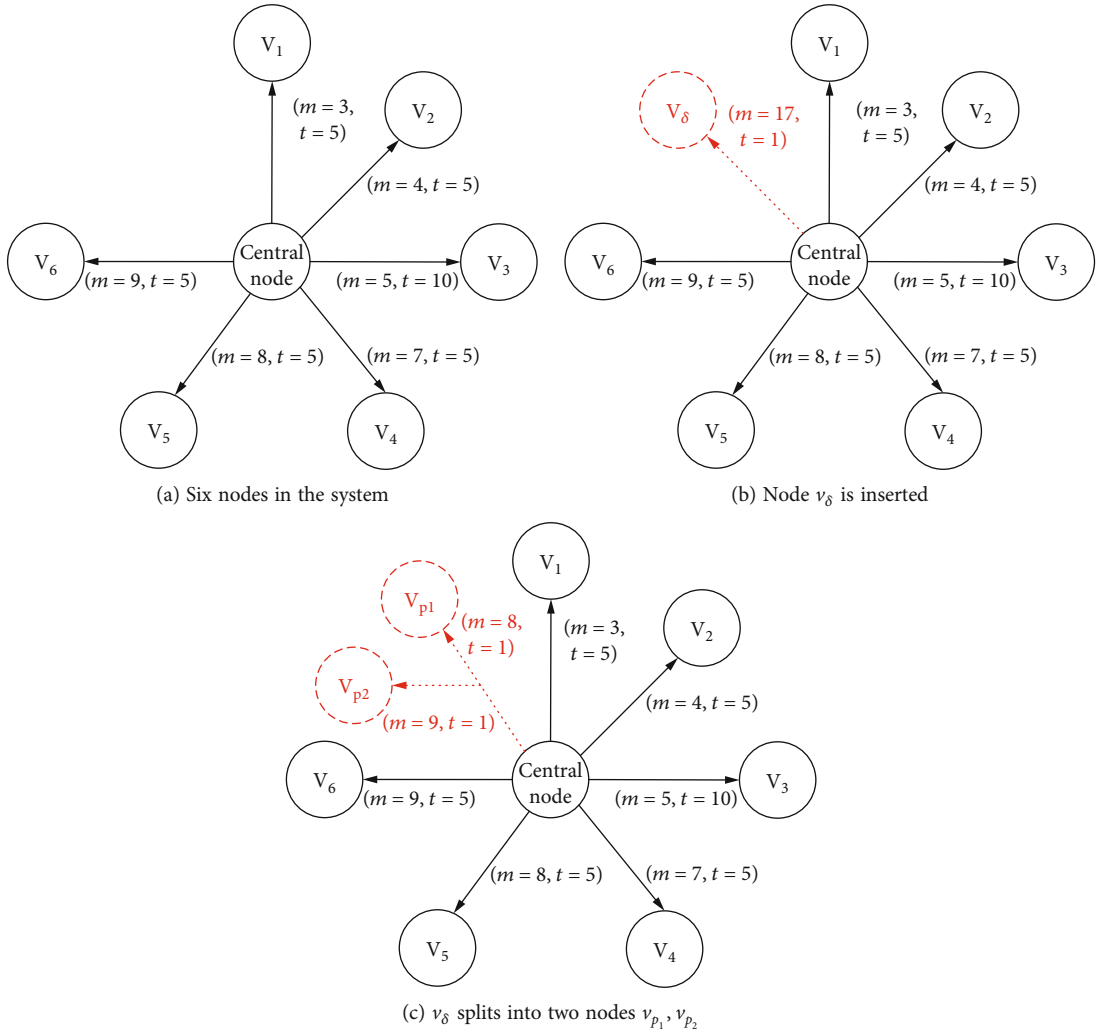
Lemma 21. *If the spare resources* R_r *are adequate*($R_r \ge t_{p_\theta} m_{p_\theta}$) *or inadequate*($R_r < t_{p_\theta} m_{p_\theta}$) *but suits* $R \ge 2t_{p_\theta} m_{k_1}$, *there are three cases:*

(i) *Case 1.* $W_{p_\theta} > \sum_{i=1}^{k_2} m_{c_i} t_{c_i}$; *there exists a splitting strategy for node* v_{p_θ} *to increase node* v_{c_i}*'s received resources*

(ii) *Case 2.* $W_{p_\theta} \le m_{c_1} \sum_{i=1}^{k_2} t_{c_i}$; *no matter how node* v_{p_θ} *splits, node* v_{c_i}*'s received resources do not change*

(iii) *Case 3.* $m_{c_1} \sum_{i=1}^{k_2} t_{c_i} < W_{p_\theta} \le \sum_{i=1}^{k_2} m_{c_i} t_{c_i}$; *there exists two real number* r *and* r_1 *satisfying the following: (a) there exists a real number* $r_1 = W_{p_\theta}/2(\sum_{i=1}^{i_1-1} t_{c_i} m_{c_i} + m_{c_{i_1}} \sum_{i=i_1}^{k_2} t_{c_i})$ *satisfying if* $m_{c_i} \le r_1$, v_{c_i}*'s received resources increase and (b) there exists a real number* r *such that if* $m_{c_i} \ge r$, v_{c_i}*'s received resources decrease*

Proof. Suppose node v_{p_θ} reports two false node $v_{p_{\theta_1}}$ and $v_{p_{\theta_2}}$ to node v_r. Consider case 1 if $W_{p_\theta} \ge \sum_{i=1}^{k_2} m_{c_i} t_{c_i}$. Let $W_{p_{\theta_1}}$ and



(a) Six nodes in the system

(b) Node v_δ is inserted

(c) v_δ splits into two nodes v_{p_1}, v_{p_2}

FIGURE 1: A centralized resource system: six nodes are connected to the central node; a new node v_δ is inserted with $m_\delta = 17$, $t_\delta = 1$; the node is split into two nodes v_{p_1}, v_{p_2} with $m_{p_1} = 8, m_{p_2} = 9$.

$W_{p_{\theta_2}}$ be the amount of resources two false nodes $v_{p_{\theta_1}}$ and $v_{p_{\theta_2}}$ receive from node v_r. If $W_{p_{\theta_1}} = \sum_{i=1}^{k_2} m_{t_{c_i}} t_{c_i}$ and $W_{p_{\theta_2}} = W_{p_\theta} - \sum_{i=1}^{k_2} m_{t_{c_i}} t_{c_i}$. Therefore, $x_{p_{\theta_1}} = m_{c_i}, x_{p_{\theta_2}} \geq 0$. $x_{p_{\theta_1}} + x_{p_{\theta_2}} \geq m_{c_i} = x_{p_\theta}$. v_{c_i}'s received resources increase. Consider case 2 if $W_{p_\theta} \leq m_{c_1} \sum_{i=1}^{k_2} t_{c_i}$, $x_{p_{\theta_1} c_i} = W_{p_{\theta_1}} / \sum_{i=1}^{k_2} t_{c_i}$, $x_{p_{\theta_2} c_i} = W_{p_{\theta_2}} / \sum_{i=1}^{k_2} t_{c_i}$, $x_{p_\theta c_i} = x_{p_{\theta_1} c_i} + x_{p_{\theta_2} c_i}$. Node v_{c_i}'s received resource does not change. Consider case 3 (a): suppose $W_{p_{\theta_1}} > W_{p_{\theta_2}}$, then $W_{p_{\theta_1}} > W_{p_\theta}/2$ and $W_{p_{\theta_2}} < W_{p_\theta}/2$. There must exists a real number $r_2 \geq r_1$ such that any node $v_{c_i}, i \in [1, k_2]$ satisfying $m_{c_i} \leq r_2$, $x_{p_{\theta_1}} = m_{c_i} = x_{p_\theta}$, and $x_{p_{\theta_2}} > 0$ under the new max-min fair allocation vector. $x_{p_{\theta_1}} + x_{p_{\theta_2}} > x_{p_\theta}$. v_{c_i}'s received resources increase. Consider case 3 (b): for node $v_{p_{\theta_1}}$'s allocation, there exists an integer $i_1^* \in [0, k_2]$ such that $x_{p_{\theta_1} c_i} = m_{c_i}, i \in [1, i_1^* - 1]$ and $x_{p_{\theta_1} c_i} < m_{c_i}, i \in [i_1^*, k_2]$. Similarly, for node $v_{p_{\theta_2}}$'s allocation, there exists an integer $i_2^* \in [0, k_2]$ such that $x_{p_{\theta_1} c_i} = m_{c_i}, i \in [1, i_2^* - 1]$ and $x_{p_{\theta_1} c_i} < m_{c_i}, i \in [i_2^*, k_2]$. If $W_{p_{\theta_1}} > W_{p_{\theta_2}}$, then $i_1^* > i_2^*$. For node v_{p_θ}'s allocation, we derive

$$W_{p_\theta} - \sum_{j=1}^{i_1^*-1} t_j m_j = \sum_{j=i_1^*}^{i^*-1} t_j m_j + x_{p_\theta c_j} \sum_{j=i^*}^{k_2} t_j. \tag{30}$$

For false nodes $v_{p_{\theta_1}}$'s and $v_{p_{\theta_2}}$'s allocation, we derive

$$W_{p_{\theta_1}} - \sum_{j=1}^{i_1^*-1} t_j m_j = x_{p_{\theta_1} c_j} \sum_{j=i_1^*}^{k_2} t_j, \tag{31}$$

$$W_{p_{\theta_2}} - \sum_{j=1}^{i_2^*-1} t_j m_j - x_{p_{\theta_2} c_j} \sum_{j=i_2^*}^{i_1^*-1} t_j = x_{p_{\theta_2} c_j} \sum_{j=i_1^*}^{k_2} t_j. \tag{32}$$

Combining Equation (30), Equation (31), and Equation (32), we derive

$$\left(x_{p_{\theta_1}c_j} + x_{p_{\theta_2}c_j}\right)\sum_{j=i_1^*}^{k_2} t_j < \sum_{j=i_1^*}^{i^*-1} t_j m_j + x_{p_\theta c_j}\sum_{j=i^*}^{k_2} t_j. \quad (33)$$

Therefore, $(x_{p_{\theta_1}c_j} + x_{p_{\theta_2}c_j}) < x_{p_\theta c_j}$; otherwise, we derive $\sum_{j=i_1^*}^{i^*-1} t_j m_j + x_{p_\theta c_j}\sum_{j=i^*}^{k_2} t_j \le x_{p_\theta c_j}\sum_{j=i_1^*}^{k_2} t_j \le (x_{p_{\theta_1}c_j} + x_{p_{\theta_2}c_j})\sum_{j=i_1^*}^{k_2} t_j$, which makes a contradiction. Thus, the lemma holds.□ □

By a similar method, the following lemma can be derived.

Lemma 22. *If the spare resources suit $R_r < t_{p_\theta} m_{p_\theta}$ and $R_r < 2t_{p_\theta} m_{k_1}$, there are three cases:*

(i) Case 1. $W_{p_\theta} > \sum_{i=1}^{k_2} m_{c_i} t_{c_i}$ or $W_{p_\theta} \le m_{c_1}\sum_{i=1}^{k_2} t_{c_i}$; node v_{c_i} can gain more resources by v_{p_θ}'s splitting strategy

(ii) Case 2. $m_{c_1}\sum_{i=1}^{k_2} t_{c_i} < W_{p_\theta} \le \sum_{i=1}^{k_2} m_{c_i} t_{c_i}$; there exists two real number r and r_1 satisfying (a) if $m_{c_i} \ge r$, v_{c_i}'s received resources decrease $(t_{p_\theta} > > \sum_{j\in[1,p_\theta-1]\bigcup[p_\theta+1,k_1]} t_j)$ or increase $(t_{p_\theta} < < \sum_{j\in[1,p_\theta-1]\bigcup[p_\theta+1,k_1]} t_j)$, and (b) there exists a real number $r_1 = W_{p_\theta}/2(\sum_{i=1}^{i_1-1} t_{c_i} m_{c_i} + m_{c_{i_1}}\sum_{i=i_1}^{k_2} t_{c_i})$ satisfying: if $m_{c_i} \le r_1$, v_{c_i}'s received resources increase

From Lemmas 21 and 22, we know that if grandparent node's resources are adequate, node's resources can not increase by parent's splitting strategy in most scenarios. However, if grandparent node's resources are inadequate, in most cases, node's resources can increase by parent's splitting strategy.

9. Numerical Examples

In this section, we present some numerical examples to illustrate these results. As shown in Figure 1(a), six nodes are connected to the central node in the system, and (m, t) (resource limitation, transfer rate) is depicted on each edge. For example, node v_1 has $m_1 = 3, t_1 = 5$. Assume the amount of all resources the central node holds is $W_c = 205$.

By Algorithm 1, the max-min allocation is listed in Table 2 where u_i is the utility of the node which is equal to the received resources x_{cv_i}; $x_{cv_i} \cdot t_i$ is the resources the central node allocates. There is no spare resource since $R = W_c - \sum_{i=1}^{6} x_{cv_i} t_i = 0$. As shown in Figure 1(b), if a new node v_δ is inserted with $m_\delta = 17, t_\delta = 1$, the max-min allocation is presented as Table 3. Nodes v_5, v_6 receive fewer resources since their resources are allocated to the new node. If the node misreports its limitation m_δ, it cannot gain more resources, and some resources would be wasted.

Table 2: The max-min fair allocation of six nodes.

Node i	m_i	t_i	x_{cv_i}	$x_{cv_i} \cdot t_i$	u_i
v_1	3	5	3	15	3
v_2	4	5	4	20	4
v_3	5	10	5	50	5
v_4	7	5	7	35	7
v_5	8	5	8	40	8
v_6	9	5	9	45	9

Table 3: The max-min allocation after v_δ is inserted.

Node i	m_i	t_i	x_{cv_i}	$x_{cv_i} \cdot t_i$	u_i
v_1	3	5	3	15	3
v_2	4	5	4	20	4
v_3	5	10	5	50	5
v_4	7	5	7	35	7
v_5	8	5	85/11 ≈ 7.727	425/11	85/11
v_6	9	5	85/11 ≈ 7.727	425/11	85/11
v_δ	17	1	85/11 ≈ 7.727	85/11	85/11

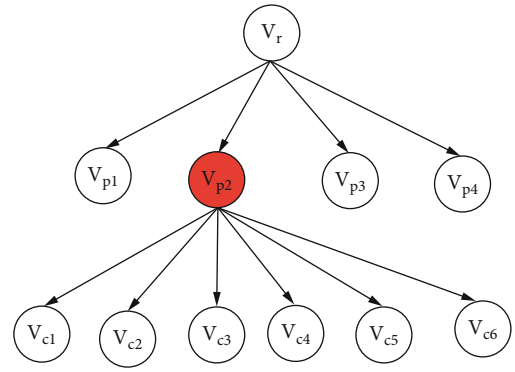


Figure 2: A tree with root v_r whose resources are passed down to descedant nodes.

In order to gain more resources, the new node splits itself into two nodes v_{p_1}, v_{p_2} such that $m_{p_1} = 8, m_{p_2} = 9$ $(m_{p_1} + m_{p_2} = m_\delta = 17)$ (as shown in Figure 1(c)). The max-min allocation is presented in Table 1; nodes v_5, v_6 receive fewer resources, and nodes v_{p_1}, v_{p_2} both receive 85/12 resources. Therefore, with the splitting strategy, the new node receives resources $x_{cv_{p_1}} + x_{cv_{p_2}} = 85/6 \approx 14.166$. Compared to the received resources in Table 3, the splitting strategy achieves $14.166/7.727 \approx 1.83$ times that of the original allocation.

Next, we present an example to illustrate splitting strategy in tree graph.

The overall amount of node v_r is $W_r = 100$. As shown in Figure 2, node v_r allocates its resources to its child $v_{pi}, i \in [1, 4]$. The resource demand m and resource transfer rate t are shown in Table 4. The amount of resource node v_{p2}'s

Table 4: The max-min allocation of node v_r's resources.

Node i	m_{pi}	t_{pi}	x_{rpi}
v_{p1}	12	2	12
v_{p2}	8	2	8
v_{p3}	20	1	20
v_{p4}	24	1	24

Table 5: The max-min allocation of node v_{p2}'s resources.

Node i	m_{ci}	t_{ci}	x_{p2ci}
v_{c1}	1	1	1
v_{c2}	2	3	2
v_{c3}	4	2	$17/5 \approx 3.4$
v_{c4}	5	1	$17/5 \approx 3.4$
v_{c5}	6	1	$17/5 \approx 3.4$
v_{c6}	8	1	$17/5 \approx 3.4$

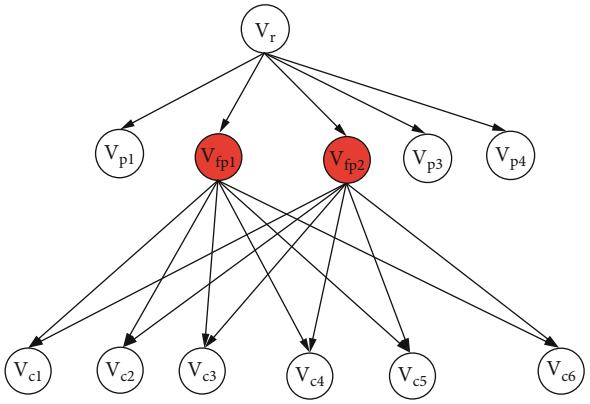


Figure 3: A tree with two false nodes v_{fp1} and v_{fp2} whose resources are passed down to descedant nodes.

Table 6: The max-min allocation of two false nodes v_{fp1} and v_{fp2}'s resources.

Node i	m_{ci}	t_{ci}	x_{fp1ci}	x_{fp2ci}
v_{c1}	1	1	1	1
v_{c2}	2	3	$11/8 \approx 1.373$	$11/8 \approx 1.373$
v_{c3}	4	2	$11/8 \approx 1.373$	$11/8 \approx 1.373$
v_{c4}	5	1	$11/8 \approx 1.373$	$11/8 \approx 1.373$
v_{c5}	6	1	$11/8 \approx 1.373$	$11/8 \approx 1.373$
v_{c6}	8	1	$11/8 \approx 1.373$	$11/8 \approx 1.373$

receives is 24. Once receiving the resources from node v_r, node v_{p2} will allocate its resources to its child $v_{ci}, i \in [1, 6]$. The max-min allocation is presented as Table 5. The amount of resource node $v_{ci}, i \in [3, 6]$ receives is fewer than their resource demand. When node v_{p2} splits itself into two false node v_{fp1} and v_{fp2} (as shown in Figure 3), the overall amount of resources node $v_{ci}, i \in [3, 6]$ receives decreases. For instance, the amount of resource node v_{p2} allocates to node v_{c3} is $x_{p2c3} = 17/5 \approx 3.4$. However, the overall resource node v_{c3} receives from two false nodes v_{fp1} and v_{fp2} equals $x_{fp1c3} + x_{fp2c3} = 22/8 \approx 2.75 < 3.4$ (as presented in Table 6).

10. Conclusion

In this paper, we study the truthfulness of the max-min fair allocation in a centralized resource system. Max-min fairness is a common adopted concept in evaluating an allocation mechanism. We propose a simple but efficient algorithm to find out the allocation. On the foundation of the algorithm, we analyze how the allocation varies when a new node is added to the system. Furthermore, we explore whether the allocation is truthful against two strategic behaviors. The misreporting strategy cannot affect the allocation while the new node can gain more resources by the splitting strategy. Hence, the max-min allocation mechanism is not truthful especially against the splitting strategy. Finally, we analyze the effect of node's splitting strategy on its child nodes on tree graph. In some situations, nodes can gain more resources despite the splitting strategy of its parents. It is still an open problem to design a fair and truthful allocation mechanism in a centralized resource system.

Data Availability

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

Disclosure

The paper is an extended version of our WASA 2020 paper "Can the Max-Min Fair Allocation Be Trustful in a Centralized Resource System?"

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Guangdong Province Key Research and Development Plan under Grant No. 2019B010136003, the National Key Research and Development Program of China (2019QY1406), the National Natural Science Foundation of China under Grant No. 61972106, the Guangdong Higher Education Innovation Group (2020KCXTD007), and the Guangzhou Higher Education Innovation Group (202032854).

References

[1] Steinhaus, "The problem of fair division," *Econometrica*, vol. 16, pp. 101–104, 1948.

[2] G. Gamow and M. Stern, *Puzzle-Math*, the Viking Press, New York, NY, USA, 1958.

[3] S. J. Brams and A. D. Taylor, "Fair division: from cake-cutting to dispute resolution," *Social Justice Research*, vol. 12, no. 2, pp. 149–162, 1999.

[4] Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia, "Truth, justice, and cake cutting," *Games & Economic Behavior*, vol. 77, no. 1, pp. 284–297, 2013.

[5] V. Menon, K. Larson, V. Menon, K. Larson, V. Menon, and K. Larson, *Deterministic, Strategyproof, and Fair Cake Cutting*, IJCAI, 2017.

[6] S. Wahlen and M. Laamanen, *Collaborative Consumption and Sharing Economies*, Routledge, 2017.

[7] L. Hurwicz, "On Informationally Decentralized Systems," *Decision and Organization: A volume in Honor of Jacob Marschak*, pp. 297–336, 1972.

[8] B. Adsul, C. S. Babu, J. Garg, R. Mehta, and M. Sohoni, *Nash Equilibria in Fisher Market*, Springer, Berlin Heidelberg, 2010.

[9] N. Chen, X. Deng, H. Zhang, and J. Zhang, "Incentive ratios of fish markets," in *International Colloquium Conference on Automata, Languages, and Programming*, pp. 464–475, Warwick, UK, 2012.

[10] Y. Cheng, X. Deng, Y. Pi, and X. Yan, *Can Bandwidth Sharing be Truthful?*, Springer, Berlin Heidelberg, 2015.

[11] Y. Cheng, X. Deng, Q. Qi, and X. Yan, "Truthfulness of a proportional sharing mechanism in resource exchange," in *International Joint Conference on Articial Intelligence (IJCAI)*, pp. 187–193, NY, USA, 2016.

[12] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1024–1039, 1991.

[13] A. Charny, *An Algorithm for Rate Allocation in a Packet*, Massachusetts Institute of Technology, 1994.

[14] P. Cretch, J. Michielsens, and J. Taeymans, *Data Network*, Prentice Hall, 2002.

[15] Z. Chen, Y. Cheng, X. Deng, Q. Qi, and X. Yan, "Agent incentives of strategic behavior in resource exchange," in *International Symposium on Algorithmic Game Theory*, pp. 227–239, L'Aquila, Italy, 2017.

[16] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, pp. 797–806, Paris, France, 2009.

[17] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "Rankclus: integrating clustering with ranking for heterogeneous information network analysis," in *Proceedings of the 12th International Conference on Extending Database Technology Advances in Database Technology - EDBT '09*, pp. 565–576, Saint Petersburg, Russia, 2009.

[18] B. Becker and H. U. Simon, "How robust is the n-cube?," *Information and Computation*, vol. 77, no. 2, pp. 162–178, 1986.

[19] E. Segal-Halevi, *Cutting a Cake with Both Good and Bad Parts*, 2017.

[20] E. Segal-Halevi and S. Nitzan, "Envy-Free Cake-Cutting among Families," 2016, http://arxiv.org/abs/1607.01517.

[21] L. Chen, S. Liu, B. Li, and B. Li, "Scheduling jobs across geo-distributed datacenters with max-min fairness," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 488–500, 2019.

[22] Y. Jia, M. Zhao, and W. Zhou, "Joint user association and eicic for max-min fairness in hetnets," *IEEE Communications Letters*, vol. 20, no. 3, pp. 546–549, 2016.

[23] J. Khamse-Ashari, G. Kesidis, I. Lambadaris, B. Urgaonkar, and Y. Zhao, "Constrained max-min fair scheduling of variable-length packetows to multiple servers," *Annals of Telecommunications*, vol. 4, pp. 1–19, 2018.

[24] T. V. Pham and A. T. Pham, "Max-min fairness and sum-rate maximization of mu-vlclocal networks," in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, San Diego, CA, USA, 2015.

[25] C. Aloui, B. Hkiri, S. Hammoudeh, and M. Shahbaz, "A multiple and partial wavelet analysis of the oil price, inflation, exchange rate, and economic growth nexus in saudi arabia," *Emerging Markets Finance & Trade*, vol. 54, no. 4, pp. 935–956, 2018.

[26] N. S. Ghumman and R. Kaur, "Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system," in *2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, TX, USA, July 2015.

[27] M. Blot, M. Cord, and N. Thome, "Max-min convolutional neural networks for imageclassication," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3678–3682, Phoenix, AZ, USA, September 2016.

[28] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *ACM Conference on Electronic Commerce*, pp. 102–111, New York, NY, USA, May 2004.

[29] N. Chen, X. Deng, and J. Zhang, "How protable are strategic behaviors in a market?," in *Algorithms – ESA 2011. ESA 2011. Lecture Notes in Computer Science, vol 6942*, C. Demetrescu and M. M. Halldórsson, Eds., pp. 106–118, Springer, Berlin, Heidelberg, 2011.

[30] T. Roughgarden and E. Tardos, "How bad is selfish routing?," *Journal of the ACM*, vol. 49, no. 2, pp. 236–259, 2002.

[31] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall Inc., 2nd edition, 1992.

[32] B. Radunovic and J. Y. le Boudec, "A Unified framework for max-min and min-max fairness with applications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1073–1083, 2007.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1604268, 10 pages
https://doi.org/10.1155/2021/1604268

Research Article

Human Origin-Destination Flow Prediction Based on Large Scale Mobile Signal Data

Qiuyang Huang,[1] **Yongjian Yang**,[1] **Yuanbo Xu**,[1] **En Wang**,[1] **and Kangning Zhu**[2]

[1]*Department of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China*
[2]*Department of Software Engineering, Jilin University, Changchun, Jilin 130012, China*

Correspondence should be addressed to Yuanbo Xu; yuanbox15@hotmail.com

Received 29 June 2021; Accepted 30 August 2021; Published 29 September 2021

Academic Editor: Wenzhong Li

The human origin-destination (OD) flow prediction is of great significance for urban safety control, stampede prevention, disease transmission control, urban planning, and many other aspects. Most of the existing methods generally divide the urban area into grids and use vehicle GPS trajectories and metrocard check-in data, combined with machine learning or deep learning models to predict human OD flow. However, these kinds of methods are challenging to capture fine-grained human mobility patterns. Moreover, these methods usually deviate from the actual human OD transfer patterns on a citywide scale due to the particularity of different datasets. To this end, in this paper, we use large-scale mobile phone signal data to achieve human OD flow prediction between the coverage of varying signal base stations. Many signal base stations are distributed in urban geographical space, collecting all the mobile phone user's location information to obtain large-scale fine-grained unbiased human OD flow data. Due to the lack of natural topology structure between base stations, this paper adopts a TGCN model combined with a graph fusion module to pretrain the dynamic population distribution prediction task. The parameters of the graph fusion module are employed to capture the different semantic information in the proposed hybrid machine learning method and finally achieve citywide human OD flow prediction. Extensive experiments on the real-world signal datasets in Changchun, China, demonstrate the effectiveness of our model.

1. Introduction

The O-D (origin-destination) flow prediction of urban residents is beneficial to grasp the dynamic trend of human mobility in urban geospatial space. It can refine and locate the flow in the face of sudden disasters, such as epidemic outbreaks, which helps to improve the vitality and responsiveness of the city. From the perspective of urban management, the OD flow between base stations is predicted and analyzed, achieving the urban population flow monitoring of base station spatial granularity, helps urban managers to study the residents' travel behavior mode, and designs and manages the urban traffic system [1, 2].

For decades, many studies on urban OD flow have been conducted to provide long-term guidance and short-term strategies for urban planning and transportation development. Existing studies usually use subway flow data and vehicle flow data as data sources, and the description of population flow is relatively simple, which cannot capture the OD flow of other travel modes. Besides, the existing studies usually focus on modeling the travel patterns from the systems which have natural topology structure (such as urban subway and road-network systems), while ignoring the spatial correlations of systems without natural topology structure [3, 4]. Figure 1 gives an example to illustrate the user, trajectory, OD pair, and signal data. When mobile phone users move around the city, mobile signal data records the visited base station sequence and time, so that we can obtain the trajectories and OD pairs of all users.

Mobile signal data provides a new solution to the problem of human OD flow prediction. In recent years, with the popularity of smartphones, the mobile signal data

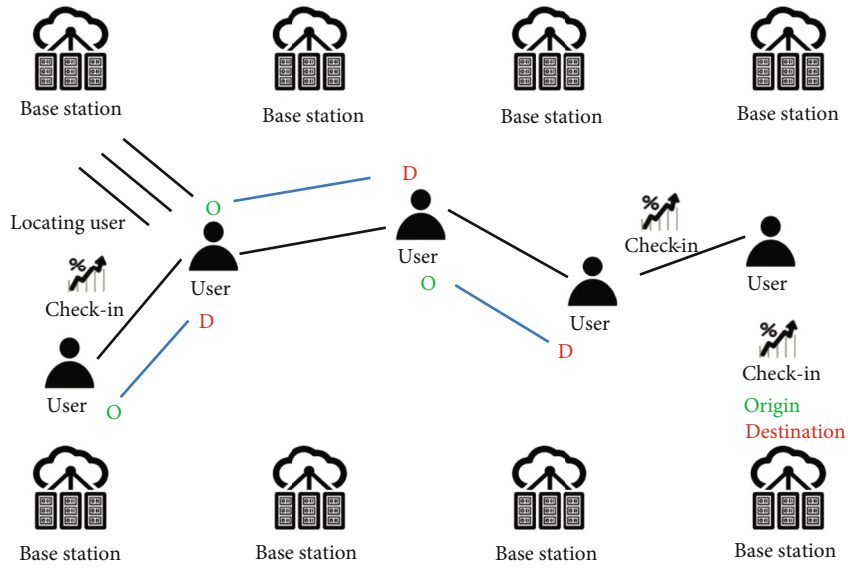


Figure 1: An example to illustrate the user, trajectory, OD pair, and signal data. When a user uses mobile phone to receive or make calls, send or receive messages, surf the Internet, and even move between the coverage of different base stations, the mobile phone will interact with the base station to obtain corresponding communication services, while the base station will passively generate mobile phone signaling data records, so that we can obtain the trajectories and OD pairs of all mobile users.

generated by the interaction between mobile devices and base stations [5] has become a data source with broad coverage, high data accuracy, and easy access. Cell phones are carried by most of the population, enabling mobile signal data to simulate all kinds of traffic modes between different OD areas, which is very suitable for traffic prediction scenarios. Besides, the OD flow between base stations is predicted and analyzed, which can achieve the monitoring of the urban population flow in base station spatial granularity, help urban manages study travel behavior mode of residents, and design and manage the urban traffic systems [6, 7].

However, urban base stations are usually installed in residential buildings and do not have natural topological structures like urban road-network or metrosystem; the OD flow links between different areas are highly dynamic. Therefore, the premise of applying the graph neural network method to extract spatial structure features in the prediction of OD flow between base stations is to find a reasonable graph structure between base stations.

Along with this line, in this paper, we hope to obtain a reasonable graph structure between base stations, a pretraining model with a graph fusion module is used to predict the number of residents, and the resulting base station graph structure is applied to the graph embedding algorithm using the idea of transfer learning to generate the embedding vectors for each base station. The embedding vectors of base stations combined with the manually extracted features (including POI and historical human mobility features) are input into the basic prediction model (such as Lasso, Random Forest, and LightGBM), to predict the OD flow between base stations. This work provides a new way to explore the effectiveness of reasonable graph structure between base stations for OD flow prediction. The contributions of this paper are summarized as follows:

(i) This paper uses large-scale mobile phone signal data to extract human mobility data, then construct training and testing dataset for OD flow prediction task. Compared with the traditional grid division-based methods, mobile signal has the advantages of large number of users, wide coverage, and fine granularity

(ii) In order to obtain the adjacency relationship between base stations in cities, this paper constructs the adjacency matrix of population flow between base stations based on the population movement times between base stations, constructs the distance adjacency matrix based on the longitude and latitude distances between base stations, and then calculates the Jaccard correlation coefficient [8] (which is used to compare the similarity and difference of POI distribution between two base stations, the greater the value is, the higher the similarity) of POI between base stations after matching urban POI data to base stations and generates the Jaccard correlation coefficient matrix. Through the weighted fusion method, the graph structure between the city base stations is finally obtained

(iii) Specifically, the base station graph structure in the pretraining model is used to train the node2vec model using the idea of transfer learning, and each base station is represented as a set of embedding vectors. Combined with the POI distribution characteristics around the base station, historical transfer traffic characteristics, and network structure characteristics, OD flow prediction is carried out, and the effectiveness of the graph embedding method is verified through experiments

2. Related Work

The main problem of human OD flow prediction is to predict the transfer flow between origin and destination areas [9]. In the early days, mathematical models such as the moving autoregressive model ARIMA [10], Kalman filter algorithm [11], and its extended algorithm were mainly used for OD traffic prediction. In recent years, various machine learning and deep learning models have been widely used in traffic OD flow prediction. These models represent the complex nonlinear relationship between different variables and provide a new prediction method [12, 13].

Gong [9] et al. propose a dual-track model OLS-DT, which learns traffic laws from two perspectives. OLS-DT can learn the steady change trend of the human travel flow and show better performance in drastic changes in the OD flow. LP Zapata et al. [14] use Bayesian inference to build a mathematical model; the Monte Carlo algorithm generates a large number of random samples; these samples will be accepted or rejected according to the Metropolis-Hasting criteria, the arithmetic average of all accepted samples as the final results. A model experiment was carried out using the transportation network in the southeastern district of Quito. Duan [15] et al. proposed a convolution-based LSTM model, which added the correlation between travel time and OD traffic and nested the city grid with roads to predict the taxi traffic between different OD areas in the city.

Generally speaking, with the development of big data computing, traffic data is becoming more and more diversified. Machine learning models and deep learning models play an increasingly important role in the current nonlinear dynamic space-time problems, and they have become an effective means to improve OD traffic prediction.

Various end-to-end neural network models have been continuously proposed and applied in appropriate business scenarios [16]. However, it is difficult for traditional deep learning methods to process non-Euclidean structured data, such as structured graph data. In practical problems, graph structure data is ubiquitous, such as bipartite graphs of user-item interactions in the recommended field, social networks, road network data in the urban field. Therefore, graph neural network (GNN) was proposed [17].

Graph neural network is a neural network used to process graph structure data [18]. This article mainly uses the graph embedding (graph embedding) method, including the graph convolutional neural network (GCN) and the graph autoencoder [19]. Graph convolutional neural networks are divided into two categories: spectrum-based methods and space-based methods. The former regards graph convolution as a filtering operation, and the data noise is removed after the convolution operation. The graph embedding method mainly learns the vector representation of the attribute graph and usually uses a set of embedding vectors to represent the nodes in the graph or the entire graph information. This method can well input graph data into machine learning algorithms. Commonly used methods include matrix factorization and random walk [20].

The graph neural network has been applied in various fields. In recommendation systems, Wang et al. [21] used the interaction between users and items to construct a bipartite graph, used graph embedding methods to obtain vector representations between users and items, and more effectively estimate user preferences for items. In the field of urban computing, Guo Shengnan et al. [22] proposed a spatiotemporal cycle convolutional network model, aiming at the short-term law of population flow, including the dependence between daily and weekly flows and unified modeling to predict urban area flow. Zhao et al. [23] extracted a time graph convolutional neural network, combined with GCN and GRU (gate recurrent unit) to capture traffic data's spatial and temporal dependence, thereby predicting traffic flow. In the field of computer vision. Fei-Fei et al. [24] used a graph convolution module to process the input of graph structure on text description generating an image, which enhanced the image generation effect.

In general, because graph neural networks can effectively capture spatial features, the research of graph neural networks has become a hot spot in various fields in recent years.

3. Framework

In this section, we introduce the detailed framework of the proposed OD flow prediction model, as shown in Figure 2. First, we construct different graphs, including geograph, transfer graph, and similarity graph, then input into a softmax network to generate the fusion graph.

After we obtain the fusion graph, we treat this mixture of different graphs as the input of the following two modules: (1) the GCN module: this module input the fusion graph into a GCN layer, then into an RNN framework (we use GRU as the node type. Note that LSTM works as well). Finally, we can achieve the node embedding with an FC layer, which includes the high-level relations (multihop neighbors in the graph) in the fusion graph e^h. (2) The node2vec module is a typical node2vec module, which aims to achieve the low level (1-hop neighbor in the graph) in the fusion graph e^l.

In addition, we extract features from the POI data and the historical human mobility data. The features include the number of different types of POIs within 300 square meters of origin and destination; OD flow in the previous K time slices; the OD flow at the same time slice in the previous day; and the average, maximum, minimum, and variance of OD flow in previous K time slices. We combine these features as e^f.

To predict the OD flow between different OD pairs, we input e^h, e^l, and e^f into the prediction model. We use different SOTA algorithms (such as the LASSO model, Random Forest model, and LightGBM model) as the prediction model to verify the effectiveness of each graph structure.

In the following section, we will give the details about our proposed model.

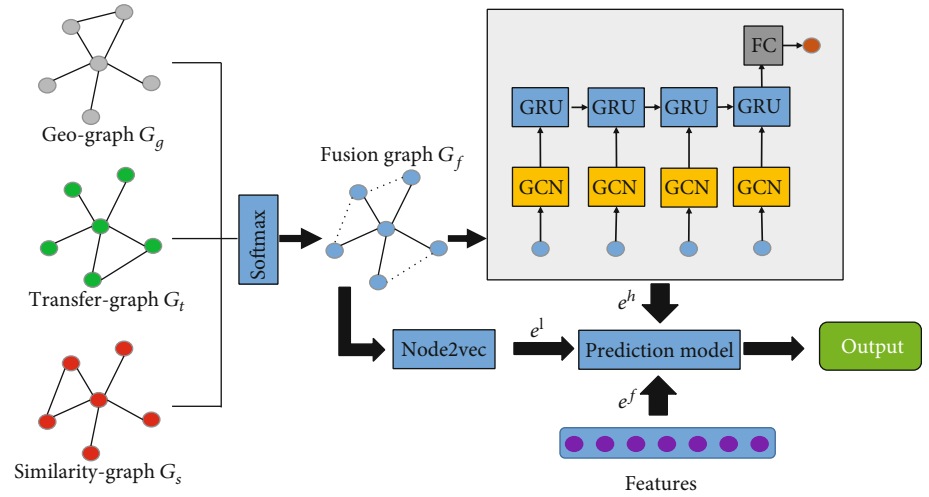


Figure 2: The framework of the proposed model.

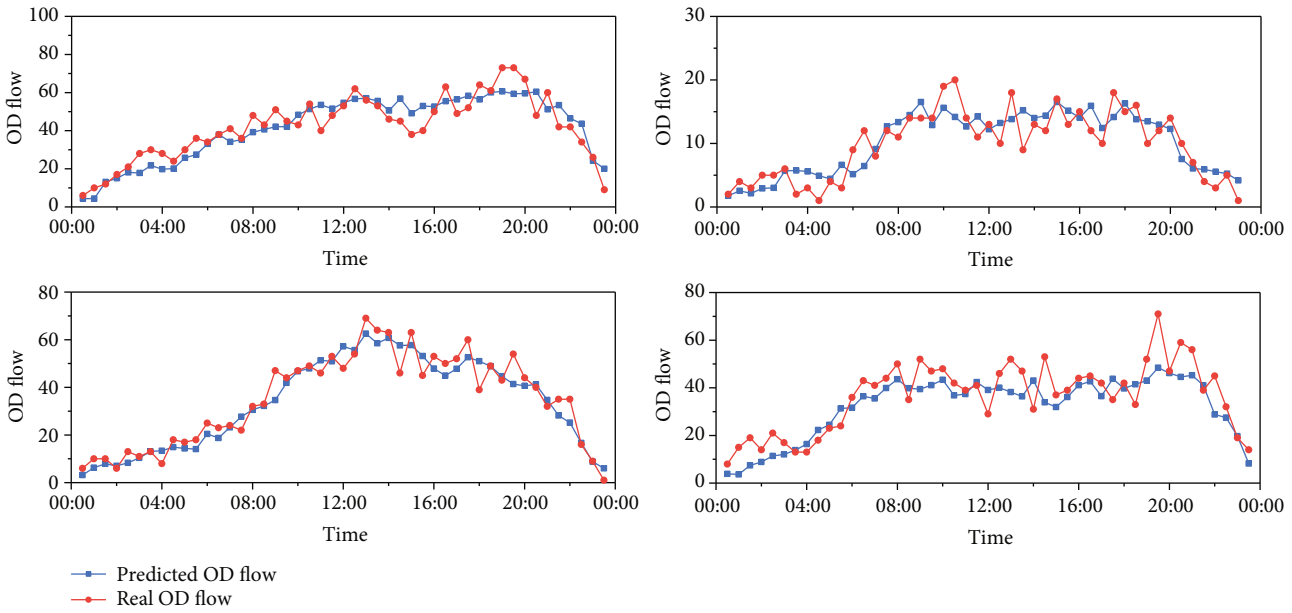


Figure 3: The OD flow time serial between different origin-destination pairs. From left-up to the right-bottom, the time interval is increasing.

4. Methodology

4.1. Definitions. This section gives detailed definitions of traffic flow, OD pairs, and several graph structures (geo-graph, transfer graph, and similarity graph). And in the last of this section, we give the problem definition.

Traffic flow: traffic flow (TF) is defined to measure the congestion level for a section in the city. Specifically, given a time point t, a section's s traffic flow can be defined as the check-in amount N_c during a period t^0. Note that the check-in amount is calculated by the signal data collected by base stations in this paper. Moreover, section s represents the circular coverage of the corresponding base station, which the center point is the longitude and latitude coordinates of the corresponding base station.

OD pairs: OD pairs are proposed to describe the users' trajectory in a fine-grained level. A user's mobile trajectory consists of different check-ins (as shown in Figure 3). An OD pair is a triplet, which is $< O, D,$ and $T >$. O stands for the origin check-in location (x_O, y_O), and D stands for the destination check-in location (x_D, y_D). T stands for the origin, destination check-in time (t_O, t_D).

Basic graph structure: we introduce three basic graph structures: (1) geograph: geograph is denoted as $G_g = \langle V_g, E_g \rangle$, where $v \in V_g$ is a section, and $s, e \in E_g$ is the geodistance between sections. (2) transfer graph: transfer graph is denoted as $G_t = \langle V_t, E_t \rangle$, where $v \in V_t$ is a section s, and $e \in E_t$ is the transfer frequency between sections, which is calculated by the historical check-in data. (3) similarity graph: similarity graph is denoted as $G_s = \langle V_s, E_s \rangle$, where $v \in V_s$ is a section s, and $e \in E_s$ is the Jaccard similarity of POI distribution between sections. With three graphs (geograph G_g, transfer graph G_t, and similarity-= graph G_s) as inputs, we employ a softmax module to build the fusion graph G_f.

Problem definition: given historical OD pairs ODs, city grid sections S, basic graph structure geograph G_g, transfer graph G_t, and section graph G_s, this model is proposed to capture all the features hidden in these data and predict the OD pairs and traffic flows after a short time period $\widehat{t}$: $\mathrm{OD}^{\widehat{t}}, \mathrm{TF}^{\widehat{t}}$.

4.2. Model Design

4.2.1. Graph Building Procedure. This section gives the details of this model, including some motivations, equations, and definitions. We calculate the distance between different sections i and sections j, and the calculation formula is as follows:

$$\mathrm{distance}_{i,j} = 2R \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\mathrm{lat}_i - \mathrm{lat}_j}{2}\right) + \cos\left(\mathrm{lat}_i\right) \cdot \cos\left(\mathrm{lat}_j\right) \cdot \sin^2\left(\frac{\mathrm{lon}_i - \mathrm{lon}_j}{2}\right)}\right). \tag{1}$$

Then, we calculate POI distributions as the initial embedding of base stations, as follows:

$$P_{si} = \left\{ p_j | \mathrm{distance}\left(s_i, p_j\right) < 300m, p_j \in P_{\mathrm{cc}} \right\}, \tag{2}$$

where P_{cc} is the POI set, and p_j is the specific POI.

If the POI distribution around two base stations is similar, it means that the coverage area of these two base stations has similar urban function expression and traffic mode. In terms of similarity of the urban function expression, this paper regards the Jaccard similarity of POI between base stations as the similarity of the urban function expression between base stations. The following formula calculates the Jaccard similarity. Jaccard similarity is used instead of cosine similarity, Pearson correlation coefficient, and other similarities because the latter two can only one-sided reflect the size of the included angle or the linear correlation between two vectors.

$$J\left(P_{s_i}, P_{s_j}\right) = \frac{\left|P_{s_i} \cap P_{s_j}\right|}{\left|P_{s_i} \cup P_{s_j}\right|} = \frac{\left|P_{s_i} \cap P_{s_j}\right|}{\left|P_{s_i}\right| + \left|P_{s_j}\right| - \left|P_{s_i} \cap P_{s_j}\right|}. \tag{3}$$

We should define the network structure between different base stations. Specifically, we treat each base station as the nodes in the graph, and the relations (recorded as adjacency matrices) are differently defined as the following, which build the several graphs:

(1) Distance adjacency matrix: we define the distance adjacency matrix as follows:

$$G_g = \langle V_g, E_g \rangle, \tag{4}$$

$$\mathrm{sign}\ (e) = \begin{cases} 1, e > 0, \\ 0, e = 0. \end{cases} \tag{5}$$

$$A_g = \begin{pmatrix} 1 & \frac{1}{dist_{0,1}} * \mathrm{sign}\ (d_{0,1}) & \frac{1}{dist_{0,2}} * \mathrm{sign}\ (d_{0,2}) & \cdots & \frac{1}{dist_{0,N-1}} * \mathrm{sign}\ (d_{0,N-1}) \\ \frac{1}{dist_{1,0}} * \mathrm{sign}\ (d_{1,0}) & 1 & \frac{1}{dist_{1,2}} * \mathrm{sign}\ (d_{1,2}) & \cdots & \frac{1}{dist_{1,N-1}} * \mathrm{sign}\ (d_{1,N-1}) \\ \frac{1}{dist_{2,0}} * \mathrm{sign}\ (d_{2,0}) & \frac{1}{dist_{2,1}} * \mathrm{sign}\ (d_{2,1}) & 1 & \cdots & \frac{1}{dist_{2,N-1}} * \mathrm{sign}\ (d_{2,N-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{dist_{N-1,0}} * \mathrm{sign}\ (d_{N-1,0}) & \cdots & \cdots & \cdots & 1 \end{pmatrix}. \tag{6}$$

This formulation defines that the closer base station pairs should achieve a close relationship.

(2) OD flow adjacency matrix: we define the OD flow adjacency matrix as follows:

$$G_t = \langle V_t, E_t \rangle, \tag{7}$$

$$A_t = \begin{pmatrix} 1 & \frac{d_{0,1}}{d_{\max}} & \frac{d_{0,2}}{d_{\max}} & \cdots & \frac{d_{0,N-1}}{d_{\max}} \\ \frac{d_{1,0}}{d_{\max}} & 1 & \frac{d_{1,2}}{d_{\max}} & \cdots & \frac{d_{1,N-1}}{d_{\max}} \\ \frac{d_{2,0}}{d_{\max}} & \frac{d_{2,1}}{d_{\max}} & 1 & \cdots & \frac{d_{2,N-1}}{d_{\max}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{d_{N-1,0}}{d_{\max}} & \cdots & \cdots & \cdots & 1 \end{pmatrix}, \tag{8}$$

where we utilize the maximum $d_{\max}$ as a regularization, and this graph considers the interactions between different base stations. Then, more OD pairs occur between them, the closer relationship they achieve in the graph.

(3) Jaccard adjacency matrix: we use Jaccard similarity to build the initial POI based base station embedding graph as follows:

$$G_s = \langle V_s, E_s \rangle, \tag{9}$$

$$A_s = \begin{pmatrix} 1 & j_{0,1} & j_{0,2} & \cdots & j_{0,N-1} \\ j_{1,0} & 1 & j_{1,2} & \cdots & j_{1,N-1} \\ j_{2,0} & j_{2,1} & 1 & \cdots & j_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ j_{N-1,0} & \cdots & \cdots & \cdots & 1 \end{pmatrix}, \tag{10}$$

where this graph denotes the similarity of different sections.

4.2.2. Model Training Procedure. The three base station graph structures are taken as input, and the adjacency matrix used for graph convolution is obtained by weighted fusion of the Softmax function. Then, the historical resident data of several time pieces are used as input to extract the spatial structure features between base stations through the graph convolution layer, and then the time correlation of data is learned through GRU. Finally, the future T-set time slice population's population resident capacity is predicted through the full connection layer. After the population resides, training task will be the figure of embedded features plus history transfer flow between base stations, base station location characteristics, base station network structure based on graph theory to extract features of standard input to the machine learning model, and forecast the OD flow between base stations.

In order to utilize different information hidden in the different graphs, we utilize the weighted fusion method to learn the final, fusion graph, which is formulated as follows:

$$\widehat{\theta}_1, \widehat{\theta}_2, \cdots, \widehat{\theta}_K = soft\max(\theta_1, \theta_2, \cdots, \theta_K), \tag{11}$$

$$\widehat{A}_i = \tilde{D}^{-\frac{1}{2}} \widehat{A}_i \tilde{D}^{-\frac{1}{2}} s.t. (1 \le i \le K), \tag{12}$$

$$\tilde{A}_f = \sum_{i=1}^{K} \widehat{\theta}_i \circ \widehat{A}_i, \tag{13}$$

where θ is the weight, and A_i denotes the ith adjacency matrix. After we achieve the fusion adjacency matrix, we input all the graphs into GCN module and do the graph convolutional action twice:

$$H^{(0)} = \mathrm{RELU}\left(\widehat{A}_f X W^{(0)}\right), \tag{14}$$

$$H^{(1)} = \sigma\left(\tilde{A}_f H^{(0)} W^{(1)}\right) = \sigma\left(\widehat{A}_f \mathrm{RELU}\left(\widehat{A}_f X W^{(0)}\right) W^{(1)}\right), \tag{15}$$

where X is the characteristic matrix of the original input 12 time slice flows, and they are calculated from the parameter matrix to achieve results from graph convolution operation. Then, the result of graph convolution is input into the GRU module to capture the temporal correlation of traffic:

$$R_t = \sigma(H_1 W_{xr} + S_{t-1} W_{sr} + b_r), \tag{16}$$

$$Z_t = \sigma(H_1 W_{xz} + S_{t-1} W_{sz} + b_z), \tag{17}$$

$$\tilde{S}_t = \tanh\left(H_1 W_{xh} + (R_t \otimes S_{t-1}) W_{ss} + b_s\right), \tag{18}$$

$$S_t = Z_t \otimes S_{t-1} + (1 - Z_t) \otimes \tilde{S}_t. \tag{19}$$

Finally, the output of the FC layer is the prediction results. This paper selects the mean square error loss function as the loss function in the pretraining model and adds regularization terms to prevent overfitting. The loss function is shown as follows, where y represents the actual value of population resident, $\hat{y}$ represents the predicted value of population resident, L_{reg} represents the regular term, and λ represents the weight parameter.

$$\mathrm{loss} = ||y - \hat{y}|| + \lambda L_{\mathrm{reg}}. \tag{20}$$

The resulting base station map structure is used for the Node2Vec algorithm to generate the embedding vector of each base station, which is input into the LightGBM model and other features constructed by feature engineering to predict OD traffic between base stations.

In this paper, feature engineering is constructed from three perspectives to make the model learn the flow transfer rules between base stations. The features are as follows: (1) historical transfer flow characteristics between base stations, (2) geographical location characteristics of base stations, and

(3) base station network structure features extracted based on graph theory.

5. Experiments

5.1. Datasets. In this paper, a large-scale signal data from July 3, 2017 to July 7, 2017 in Changchun is obtained to train and verify the performance of the proposed model. The dataset contains 200 million cell phone users and a total of 49,716,815 signal records. The first 4 days is used for training, and the last is used for testing.

5.2. Prediction Model Selection. We select the following three models as the prediction model, respectively, to compare the effects of different graph structures on the performance of the proposed model:

Lasso [25]: a compression estimation method is based on the idea of reducing the variable set (order reduction). By constructing a penalty function, it can compress the coefficients of variables and make some regression coefficients that become 0, so as to achieve the purpose of variable selection.

Random Forest [26]: random forests have been one of the successful ensemble algorithms in machine learning. The basic idea is to construct many random trees individually and make predictions based on an average of their predictions. The great successes have attracted much attention on the consistency of random forests, mostly focusing on regression.

LightGBM [27]: LightGBM excludes a significant proportion of data instances with small gradients and only use the rest to estimate the information gain. It can obtain quite an accurate estimation of the information gain with a much smaller data size and bundle mutually exclusive features (i.e., they rarely take nonzero values simultaneously), to reduce the number of features.

To verify the effectiveness of each graph structure, we test the accuracy of the prediction results by using control experiments, that is using different graph combinations (without graph embedding, with G_g, with G_t, with G_s, with fusion of all graphs) as the input of the prediction model. The results are shown in Table 1.

5.3. Experimental Results

5.3.1. Prediction Accuracy Evaluation. The prediction results are shown in Table 1. To avoid the difference of the prediction results that is too tiny, we filter out the OD flow whose predicted value and the ground truth are both 0, then calculate the loss and RMSE as shown in Table 1. We can see that LightGBM achieves better performance in both of these three selected prediction models, which can better fit the nonlinear relationship in the data. The lasso regression model has poor performance due to its simplicity. By using LightGBM as the prediction model, with fusion of all the graph structures, the performances of loss and RMSE gain 67.72% and 64.03%, respectively, than without embedding. With only one graph structure as embedding feature, transfer graph G_t makes more significant performance improvements than geograph G_g and similarity graph G_s. Without embedding as input achieves the worst performance in both of the prediction models.

In summary, the graph fusion embedding feature can significantly improve the performance of OD flow prediction, and transfer graph G_t plays a more important role than geograph G_g and similarity graph G_s.

Table 1: Experimental result comparison.

	Lasso		Random Forest		LightGBM	
	Loss	RMSE	Loss	RMSE	Loss	RMSE
No embedding	19.74	25.12	18.82	23.45	18.74	23.49
With G_g	14.96	19.98	14.70	18.66	14.01	18.17
With G_t	9.72	12.27	7.80	10.38	7.72	10.06
With G_s	12.77	15.83	12.73	15.75	12.69	15.43
With fusion	7.45	9.54	6.51	9.37	6.05	8.45

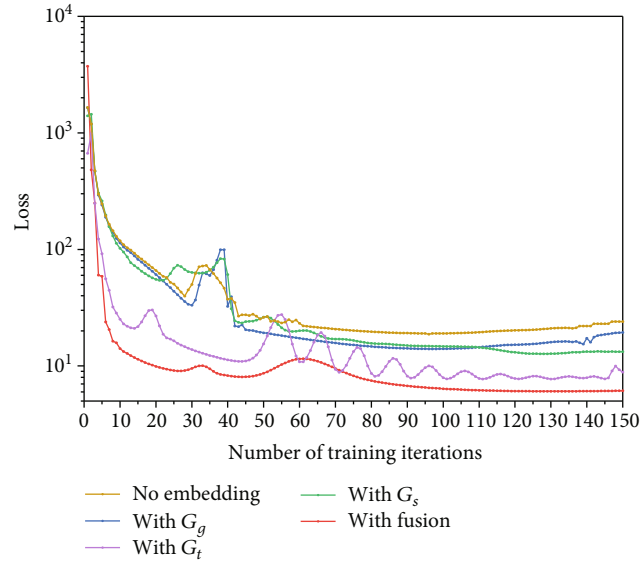


Figure 4: The convergence of the proposed model, with LightGBM as the prediction model, under different graph structure combinations.

Table 2: The training time comparison.

	Using LightGBM as the prediction model	
	Number of iterations with minimum loss	Training time (hours)
No embedding	95	1.97
With G_g	96	2.28
With G_t	111	2.19
With G_s	128	2.37
With fusion	129	3.15

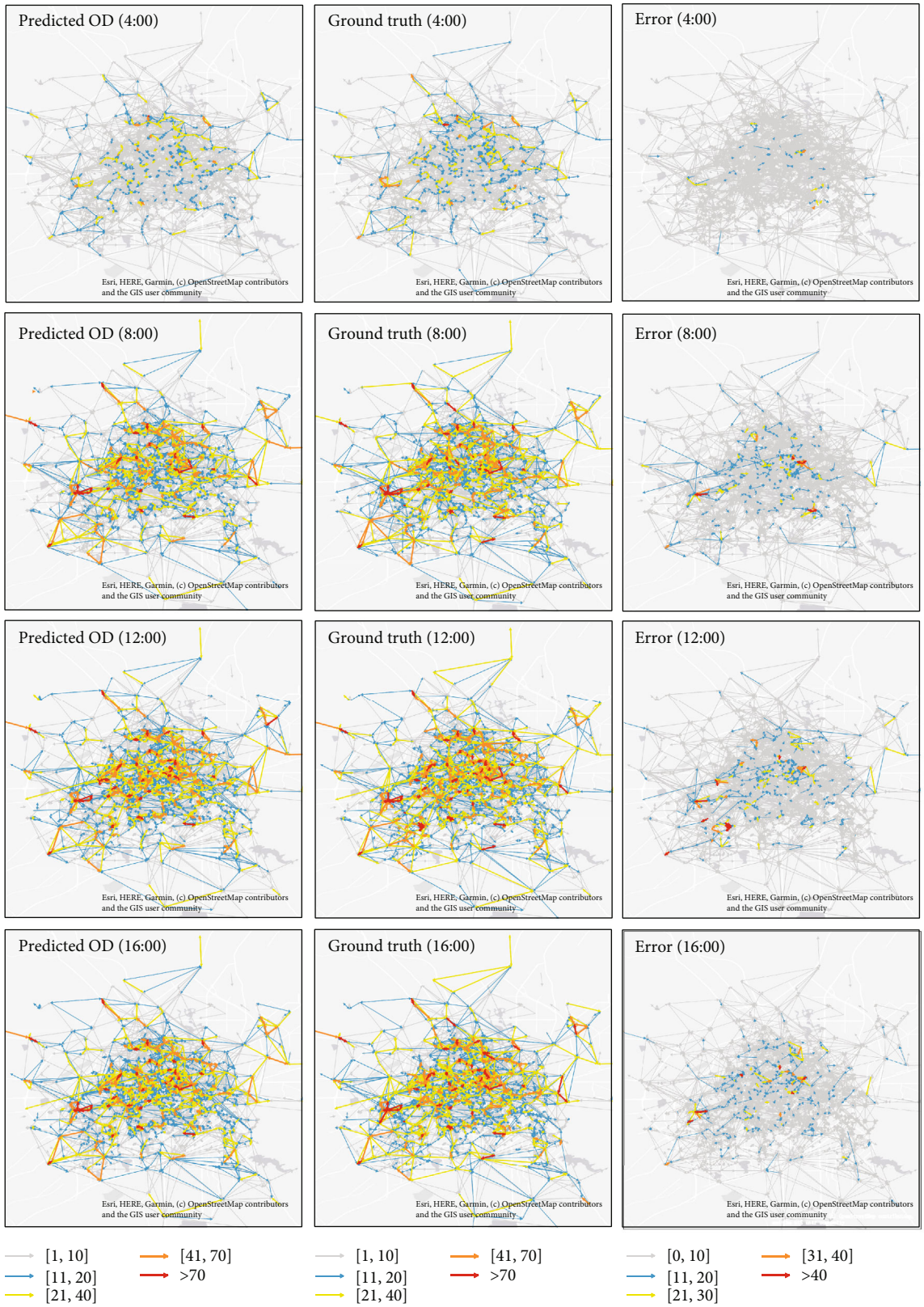


Figure 5: OD flow prediction results in different time periods. Left column predicted OD flow. Middle column is the ground truth. Right column is the absolute error between the prediction results and the ground truth.

5.3.2. Computation Time and Convergence. In this paper, we implement the proposed framework by using python 3.6, tensorflow-1.12.0. The hardware environment is with 16 GB RAM, 3.20 GHz Intel(R) Core (TM) i7-8700 CPU). Under above experimental environment, we select LightGBM as prediction model, to discuss the computation time and convergence under different graph structure combinations, and similar conclusions can be easily extended to the other two prediction models. As shown in Figure 4, the loss of each situation decreased significantly in the first 30 training iterations. Among them, the loss of fusing all graph structures decreases the fastest, followed by using transfer graph G_t, and the loss of without graph embedding decreases the slowest and achieves the worst performance.

We set maximum training iterations as 150, and the number of iterations with minimum loss and the training time (hours) is presented in Table 2. We can find that fusing all graph structures costs more training time (3.15 hours).

5.3.3. Ablation Evaluation. In this section, we separate the time periods into different time pieces. So, we could explore the effect of different OD pairs on our proposed model. Note that the longer OD pair time could lead to a fierce shift in the performance, as shown in Figure 3:

From the results, we can see that with the increasing of the time interval between OD pairs, the performance of our proposed model is decreasing. However, our proposed model still captures OD pairs' tendency, which proves the robustness of our proposed model.

5.3.4. City Scale OD Pair Prediction. In this section, we want to explore the city scale OD pair prediction performance. Specifically, we give the comparison between predict results and ground truth. The results are shown in Figure 5.

The left column is the prediction, the middle column is the ground truth, and the right column is the absolute errors. Each row is a time period (0 : 00-6 : 00, 6 : 00-12 : 00, 12 : 00-18 : 00, 18 : 00-24 : 00). From the result, we can see that our proposed model could capture the city scale OD pair characters and give a proper evaluation of different sections. Specifically, we can see from the right column that our proposed model achieves the best performance in the first time period (0 : 00-6 : 00). The reason is that at this time, the users' action pattern is simple but predictable. Usually, they are works, doctors, and city cleaners. So, the prediction accuracy is quite high. Note that there are several sections where the prediction performance is not satisfying in the following three time periods. Specifically, take deep into this scenario, we can see that the low prediction accuracy sections have some similar characters: first, there sections' traffic flow are always huge and unstable. Some sections are CBDs where locates at the city's center. So, the flow prediction is difficult for our proposed model and all the prediction models because there are too many reasons that could affect these sections' traffic flow.

6. Conclusion

This paper uses large-scale mobile phone signal data to achieve human OD flow prediction between the coverage of varying signal cell towers. Many signal cell towers are distributed in urban geographical space, collecting all the cell phone user's location information to obtain large-scale fine-grained unbiased human OD flow data. Extensive evaluation proves the proposed model's superior performance over some SOTA methods.

In the future, we plan to combine this model with other aspects to build a smart city, such as functional zone division, road traffic congestion monitoring, and other applications. We believe this work could be basic work for other high-level algorithms.

Data Availability

The data used to support the findings of this study have not been made available because the data also forms part of an ongoing study.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundations of China under Grant No. 61772230, No. 61976102, No. U19A2065, and No. 61972450, Natural Science Foundation of China for Young Scholars No. 61702215 and No. 62002132, China Postdoctoral Science Foundation No. 2020M681040, Changchun Science and Technology Development Project No. 18DY005, National Defense Science and Technology Key Laboratory Fund Project No. 61421010418, Science Foundation of Jilin Province No. 20190201022JC, and China National Postdoctoral Program for Innovative Talents No. BX20180140.

References

[1] Y. Yang, Y. Xu, J. Han, E. Wang, W. Chen, and L. Yue, "Efficient traffic congestion estimation using multiple spatio-temporal properties," *Neurocomputing*, vol. 267, pp. 344–353, 2017.

[2] H. Hu, Z. Jiang, Y. Zhao, Y. Zhang, H. Wang, and W. Wang, "Network representation learning-enhanced multisource information fusion model for POI recommendation in smart city," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9539–9548, 2021.

[3] Q. Huang, Y. Yang, Y. Xu, F. Yang, Z. Yuan, and Y. Sun, "City-wide road-network traffic monitoring using large-scale mobile signaling data," *Neurocomputing*, vol. 444, pp. 136–146, 2021.

[4] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7919–7927, 2020.

[5] J. Yang, B. Guo, Z. Wang, and Y. Ma, "Hierarchical prediction based on Network-Representation-Learning-Enhanced clustering for bike-sharing system in smart city," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6416–6424, 2021.

[6] Y. Xu, Y. Yang, E. Wang et al., "Neural serendipity recommendation," *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 4, pp. 1–25, 2020.

[7] Y. Xu, Y. Yang, J. Han, E. Wang, J. Ming, and H. Xiong, "Slanderous user detection with modified recurrent neural networks in recommender system," *Information Sciences*, vol. 505, pp. 265–281, 2019.

[8] K. Song, J. Min, G. Lee, S. C. Shin, and Y. S. Kim, "An improvement of plagiarized area detection system using jaccard correlation coefficient distance algorithm," *Computer Science and Information Technology*, vol. 3, no. 3, pp. 76–80, 2015.

[9] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, "Online Spatio-Temporal Crowd Flow Distribution Prediction for Complex Metro System," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2020.

[10] H. Yang, X. Li, W. Qiang, Y. Zhao, W. Zhang, and C. Tang, "A network traffic forecasting method based on SA optimized ARIMA-BP neural network," *Computer Networks*, vol. 193, article 108102, 2021.

[11] S. Kim, V. M. Deshpande, and R. Bhattacharya, "Robust kalman filtering with probabilistic uncertainty in system parameters," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 295–300, 2021.

[12] Y. Xu, E. Wang, Y. Yang, and Y. Chang, "A unified collaborative representation learning for neural-network based recommendersystems," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[13] Y. Xu, Y. Yang, E. Wang, F. Zhuang, and H. Xiong, "Detect professional malicious user with metric learning in recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[14] L. P. Zapata, M. Flores, V. M. Larios-Rosillo, R. Maciel, and E. A. Antunez, "Estimation of people flow in´ public transportation network through the origin-destination problem for the south-eastern corridor of quito city in the smart cities context," in *2019 IEEE International Smart Cities Conference, ISC2 2019*, pp. 181–186, Casablanca, Morocco, 2019.

[15] Z. Duan, K. Zhang, Z. Chen et al., "Prediction of city-scale dynamic taxi origin-destination flows using a hybrid deep neural network combined with travel time," *IEEE Access*, vol. 7, pp. 127816–127832, 2019.

[16] S. Meng, S. Sun, and B. Yang, "Traffic flow prediction with conv-sae," in *Proceedings of the 2020 the 7th International Conference on Automation and Logistics (ICAL)*, pp. 90–93, Beijing, China, 2020.

[17] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[18] J. Zhou, G. Cui, S. Hu et al., "Graph neural networks: a review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[19] Z. Zhu, R. Li, M. Shan et al., "TDP: personalized taxi demand prediction based on heterogeneous graph embedding," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*, pp. 1177–1180, Paris, France, 2019.

[20] L. Grad-Gyenge, A. Kiss, and P. Filzmoser, "Graph embedding based recommendation techniques on the knowledge graph," in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP 2017*, pp. 354–359, Bratislava, Slovakia, 2017.

[21] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*, pp. 165–174, Paris, France, 2019.

[22] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3913–3926, 2019.

[23] L. Zhao, Y. Song, C. Zhang et al., "T-GCN: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[24] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pp. 1219–1228, Salt Lake City, UT, USA, 2018.

[25] C. Han, J. Shimamura, T. Takahashi et al., "Real-time detection of malware activities by analyzing darknet traffic using graphical lasso," in *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2019*, pp. 144–151, Rotorua, New Zealand, 2019.

[26] W. Gao and Z. Zhou, "Towards convergence rate analysis of random forests for classification," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, pp. 9300–9311, Vancouver, Canada, 2020.

[27] G. Ke, Q. Meng, T. Finley et al., "Lightgbm: a highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 3146–3154, Long Beach, CA, USA, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5586029, 15 pages
https://doi.org/10.1155/2021/5586029

Research Article

Doppler Spread Estimation Based on Machine Learning for an OFDM System

Eunchul Yoon,[1] **Soonbum Kwon**,[1] **Unil Yun**,[2] **and Sun-Yong Kim**[1]

[1]*Department of Electronic Eng. at Konkuk University, Seoul, Republic of Korea*
[2]*Department of Computer Eng., Sejong University, Seoul, Republic of Korea*

Correspondence should be addressed to Sun-Yong Kim; kimsy@konkuk.ac.kr

Received 7 January 2021; Revised 7 July 2021; Accepted 30 August 2021; Published 22 September 2021

Academic Editor: Wenzhong Li

In this paper, we propose a Doppler spread estimation approach based on machine learning for an OFDM system. We present a carefully designed neural network architecture to achieve good performance in a mixed-channel scenario in which channel characteristic variables such as Rician K factor, azimuth angle of arrival (AOA) width, mean direction of azimuth AOA, and channel estimation errors are randomly generated. When preprocessing the channel state information (CSI) collected under the mixed-channel scenario, we propose averaged power spectral density (PSD) sequence as high-quality training data in machine learning for Doppler spread estimation. We detail intermediate mathematical derivatives of the machine learning process, making it easy to graft the derived results into other wireless communication technologies. Through simulation, we show that the machine learning approach using the averaged PSD sequence as training data outperforms the other machine learning approach using the channel frequency response (CFR) sequence as training data and two other existing Doppler estimation approaches.

1. Introduction

Information about Doppler spread can be used for handoff, adaptive modulation, equalization, power control, etc., in wireless communication systems. Various Doppler spread estimation approaches have been studied for isotropic scattering and Rayleigh fading channels [1–6]. The Doppler spread estimation approaches of [5, 6] in single-antenna systems have been shown to outperform previous Doppler spread estimation approaches in isotropic scattering and Rayleigh fading channels. However, the performances of previous Doppler spread estimation approaches designed for isotropic scattering and Rayleigh fading channels degrade when the channels are shaped by nonisotropic scattering and line-of-sight (LOS) channel components. Therefore, it is still worthwhile to develop powerful new techniques for nonisotropic scattering and Rician fading channels for single-antenna systems. Meanwhile, machine learning has been the focus of extensive research in recent years because of its empirical success in various fields [7–24]. First of all, machine learning has been used for visual object recognition and speech recognition [7, 8]. Machine learning-based approaches have been successfully applied to wireless communication systems [9–18]. Recently, efforts have been made to apply machine learning to estimate the angle of arrival (AoA) [19–22] and apply it to indoor positioning [23, 24]. In [25], machine learning was used to improve the performance of human detection and activity classification based on the information of micro-Doppler signatures, i.e., Fourier transforms of the Doppler radar data measured under line-of-sight (LOS) conditions. In [26], machine learning was used to improve the performance of vehicle collision avoidance services based on information from the Doppler profiles, i.e., spectral representations of temporal non-line-of-sight (NLOS) Doppler energy. In [27], machine learning was used to estimate Doppler spread based on channel state information (CSI) by a high-speed train (HST) system. The channel considered in [27] differs from a typical mobile radio communication channel in that it features a LOS component formed based on the position of the HTS moving on the same track with respect to adjacent radio head units. To the best of our knowledge, machine learning has not been used for

Doppler spread estimation in the context of general mobile radio communication channels. This led to the study of applying machine learning to Doppler spread estimation for mobile radio communication channels in this paper.

Machine learning can work efficiently even when the system model is unknown, or the parameters cannot be accurately estimated. Doppler spread estimation techniques designed for specific channel conditions do not work well in real-world channel environments where channel characteristic variables such as Rician K factor, azimuth angle of arrival (AOA) width, average azimuth AOA, and channel estimation error are arbitrarily generated. Therefore, it is interesting to see if additional gains in performance can be obtained by applying a machine learning approach to Doppler spread estimation, especially when the channel characteristic variables are randomly generated. In [28], it was mentioned that machine learning requires a very large number of training data to effectively train the weights of a neural network for accurate classification. In other words, machine learning will not perform well if the amount of training data is not sufficient. However, it is often difficult to obtain a very large number of training data in real mobile radio communication systems. Given a limited number of training data, preprocessing the training data into high-quality training data by using feature selection and feature extraction can improve the performance of machine learning [29, 30]. This motivated our study to find out how to preprocess the collected CSI into high-quality training data to improve the machine learning performance for Doppler spread estimation.

In this paper, we propose a Doppler spread estimation approach based on machine learning for an OFDM system. We present a carefully designed neural network architecture to achieve good performance in a mixed-channel scenario in which channel characteristic variables such as Rician K factor, azimuth AOA width, mean direction of azimuth AOA, and channel estimation errors are randomly generated. When preprocessing the CSI collected under the mixed-channel scenario, we propose averaged power spectral density (PSD) sequence as high-quality training data in machine learning for Doppler spread estimation. We detail intermediate mathematical derivatives of the machine learning process, making it easy to graft the derived results into other wireless communication technologies. Through simulation, we show that the machine learning approach using averaged PSD sequence as training data outperforms the other machine learning approach using channel frequency response (CFR) sequence as training data and two other existing Doppler estimation approaches. The main contributions of this paper are summarized as follows: (1) This paper applies machine learning to Doppler spread estimation in the context of mobile radio communication channels in a mixed-channel scenario. (2) The machine learning process detailed in this paper can be easily applied to other wireless communication technologies as well.

The rest of this paper is organized as follows. Section 2 describes the system model. Section 3 proposes two machine learning approaches and presents comprehensive algorithms for grafting machine learning into Doppler spread estimation. Section 4 presents the simulation results and compares the performance of the proposed and conventional approaches. Finally, Section 5 provides concluding remarks.

2. System Model

We consider an OFDM system with FFT size N_F. We assume that the number of OFDM symbols required to compute the PSD function is N_{data}, and the number of test user equipments (TUEs) required to collect training data for machine learning is N_{TUE}. The channel impulse response of the m-th TUE for $m = 1, 2, \cdots, N_{\text{TUE}}$ is modeled as a tapped delay line with L taps,

$$\mathbf{h}^{(m)}[n] = \left[h_0^{(m)}[n] h_1^{(m)}[n] \cdots h_{L-1}^{(m)}[n] \right], \tag{1}$$

for $n = 0, 1, \cdots, N_{\text{data}} - 1$, where n denotes the OFDM symbol index and $h_l^{(m)}[n]$ denotes the coefficient of the l-th channel path. To consider the effects of nonisotropic scattering and LOS channel components on Doppler spread estimation, we model the channel coefficients as in [1] by

$$h_l^{(m)}[n] = \sqrt{\frac{\Omega_{h,l}}{K_r^{(m)} + 1}} h_{l,a}^{(m)}[n] + \sqrt{\frac{K_r^{(m)} \Omega_{h,l}}{K_r^{(m)} + 1}} h_{l,b}^{(m)}[n], \tag{2}$$

where $h_{l,a}^{(m)}[n]$ and $h_{l,b}^{(m)}[n]$ denote non-LOS and LOS components, respectively, $K_r^{(m)}$ denotes the Rician K-factor defined on a linear scale meaning the ratio of LOS component power to non-LOS (or fluctuating) component power, and $\Omega_{h,l}$ denotes the power delay profile. For Rician fading, most estimated K-factor values are less than 3 dB in urban areas [31]. For the simulation, we assume an exponential power delay profile where $\Omega_{h,l}$ for $l = 0, \cdots, L-1$ is given by

$$\Omega_{h,l} = \frac{\rho^l}{\sum_{l=0}^{L-1} \rho^l}, \tag{3}$$

with $\rho = 0.8$. The autocorrelation function of $h_{l,a}^{(m)}[n]$ was presented in [1, 5, 32] as

$$r_{h_{l,a}^{(m)}}[n] = \frac{I_0\left(\sqrt{\left(\left(\kappa^{(m)} \right)^2 - \left(2\pi f_D^{(m)} T_S n \right)^2 + j4\pi \kappa^{(m)} f_D^{(m)} T_S n \cos \alpha^{(m)} \right)} \right)}{I_0\left(\kappa^{(m)} \right)}, \tag{4}$$

where $f_{\text{D}}^{(m)}$ denotes the Doppler spread, $\kappa^{(m)}$ denotes the azimuth AOA width factor which controls the width of the azimuth AOA, $\alpha^{(m)}$ denotes the mean direction of the azimuth AOA, T_S denotes the OFDM symbol period, and $I_0(\cdot)$ denotes the zero-th order modified Bessel function of the first kind. The l-th Rician channel component, $h_{l,b}[n]$, can be written as

$$h_{l,b}^{(m)}[n] = e^{-j2\pi f_D^{(m)} T_S n \cos \alpha_0^{(m)} + j\phi_0^{(m)}}, \tag{5}$$

where $\alpha_0^{(m)}$ is the parameter controlling the azimuth AOA direction of the LOS component and $\phi_0^{(m)}$ is the parameter representing the phase of the LOS component with a uniform distribution between 0 and 2π. The autocorrelation function of $h_l^{(m)}[n]$ was presented in [5, 32] as

$$r_{h_l^{(m)}}[n] = \frac{\Omega_{h,l}}{K_r^{(m)}+1} r_{h_{l,a}^{(m)}}[n] + \frac{K_r^{(m)}\Omega_{h,l}}{K_r^{(m)}+1} e^{-j2\pi f_D^{(m)} T_S n \cos \alpha_0^{(m)}}. \quad (6)$$

The CFR coefficient over the k-th subcarrier is given by the discrete Fourier transform (DFT) of $\mathbf{h}^{(m)}[n]$,

$$H_k^{(m)}[n] = \sum_{l=0}^{L-1} h_l^{(m)}[n] e^{-j(2\pi/N_F)kl}, \quad (7)$$

for $k = 0, 1, \cdots, N_F - 1$. The OFDM demodulated signal over the k-th subcarrier of the n-th OFDM symbol for the m-th TUE can be written as

$$Y_k^{(m)}[n] = H_k^{(m)}[n] X_k^{(m)}[n] + W_k^{(m)}[n], \quad (8)$$

where $X_k^{(m)}[n]$ denotes the transmitted symbol and $W_k^{(m)}[n]$ denotes a zero-mean circularly symmetric complex Gaussian noise with variance σ^2. SNR is defined as $1/\sigma^2$ assuming that the transmitted symbol has unit average power. If the number of pilot symbols in the OFDM block is N_P, the CFR coefficient over the subcarrier of the p-th pilot symbol for $p = 0, 1, \cdots, N_P - 1$ can be estimated by the least square detection method in [33] as

$$\widehat{H}_{n_p}^{(m)}[n] = \frac{Y_{n_p}^{(m)}[n]}{X_{n_p}^{(m)}[n]}, \quad (9)$$

where n_P denotes the subcarrier index of the p-th pilot symbol. We use the minimum mean square error-based channel interpolation method in [34] to estimate the CFR coefficients, $\{\widehat{H}_k^{(m)}[n]\}_{0\le n\le N_{\text{data}}-1, 0\le k\le N_F-1}$. By using this channel estimation method, the channel estimation error increases as the SNR decreases. The ideal PSD function of the channel can be obtained as in [35] by taking the Fourier transform of $r_{h_l^{(m)}}[n]$,

$$S_{\text{ideal},l}^{(m)}[u] = \sum_{n=-\infty}^{\infty} r_{h_l^{(m)}}[n] e^{-j(2\pi/N_{\text{data}})un}, \quad (10)$$

for $u = 0, 1, \cdots, N_{\text{data}}/2 - 1$. The Doppler spread index is defined by

$$u_D^{(m)} = \frac{f_D^{(m)}}{\Delta f} = f_D^{(m)} T, \quad (11)$$

where $T(= N_{\text{data}} T_S)$ denotes the period of collecting information of N_{data} consecutive OFDM symbols and $f_D^{(m)}(= 1/T)$ denotes the frequency resolution required to determine the Doppler spread index with the Doppler spread. Since base stations (BSs) in macrocells generally suffer from severe nonisotropic scattering, $S_{\text{ideal},l}^{(m)}[u]$ has its maximum quantity at $u_{\max,l}^{(m)} = u_D^{(m)} \cos \alpha^{(m)}$ [5]. For severe nonisotropic scattering, it is reasonable to assume $\alpha_0^{(m)} = \alpha^{(m)}$ for all $\{h_l^{(m)}[n]\}_{l=0}^{L-1}$ [36]. The Doppler spread is found by transforming $u_{\max,l}^{(m)}$ to $f_{D,l}^{(m)}$ based on $f_{D,l}^{(m)} = u_{\max,l}^{(m)}/T/\cos \alpha^{(m)}$ for $l = 0, 1, \cdots, L-1$ and finding $f_D^{(m)} = \max_{l\in\{0,\cdots,L-1\}} f_{D,l}^{(m)}$ [5]. For severe nonisotropic scattering, the angle spread $\theta^{(m)}$ is mostly less than 30°, and for very severe nonisotropic scattering, it is less than 10° [5]. If $\kappa^{(m)}$ is not too small, $\kappa^{(m)}$ can be approximated as $\kappa^{(m)} = (360°/\theta^{(m)}/\pi)^2$ [37]. We assume that the transmitted signal undergoes severe nonisotropic scattering and that $\theta^{(m)}$ has a uniform distribution between 10° and 30°. Although there are several ways to estimate the PSD function [38], periodogram-based nonparametric techniques are often used because of its simplicity. Using the periodogram-based nonparametric technique, multiple PSD functions can be computed with multiple series of CFR coefficients; the k-th PSD function computed using a series of the k-th estimated CFR coefficient, $\{\widehat{H}_k^{(m)}[n]\}_{0\le n\le N_{\text{data}}-1}$, can be written as

$$S_k^{(m)}[u] = \frac{1}{N_{\text{data}}} \left| \sum_{n=0}^{N_{\text{data}}-1} H\wedge_k^{(m)}[n] e^{-j(2\pi/N_{\text{data}})un} \right|^2, \quad (12)$$

for $k = 0, 1, \cdots, N_F - 1$ and $u = 0, 1, \cdots, N_{\text{data}}/2 - 1$. Because of channel estimation error and the lack of channel coefficients used to compute the PSD function, the accuracy of the PSD function decreases. With an inaccurate PSD function, the maximum value of $S_k^{(m)}[u]$ given in (12) does not always occur at the ideal Doppler spread index $u = u_D^{(m)} \cos \alpha^{(m)}$. Since $f_D^{(m)} = \Delta f^{(m)} u_D^{(m)} \cos \alpha^{(m)}$, a slight deviation in the value of $u_D^{(m)}$ results in a $\Delta f^{(m)} \cos \alpha^{(m)}$ times greater deviation in the $f_D^{(m)}$ value. Therefore, it is important to set $\Delta f^{(m)}$ to a small value to improve the performance of the nonparametric Doppler spread estimation approach. In order for $\Delta f^{(m)}$ to be small with T_S fixed, N_{data} should be larger because $\Delta f^{(m)} = 1/(N_{\text{data}} T_S)$. The sampling theorem states that in the time dimension, the channel sampling rate must be at least twice the Doppler spread, i.e., $1/T_S \ge 2 f_D^{(m)}$. A good rule of thumb for finding the maximum of $f_D^{(m)}$ that can be estimated by a Doppler spread estimation approach is to assume that the maximum of $f_D^{(m)} T_S$ lies between 1/12 and 1/6 ([35], pp. 845–846). We set the maximum value of $f_D^{(m)} T_S$ to 0.1. Therefore, we assume that $0 \le f_D^{(m)} \le 0.1/T_S$, or equivalently, $0 \le u_D^{(m)} \le N_{\text{data}}/10$.

3. Proposed Doppler Spread Estimation

Figure 1 shows the block diagram of the proposed neural network structure to realize forward signal propagation in machine learning. The proposed neural network structure

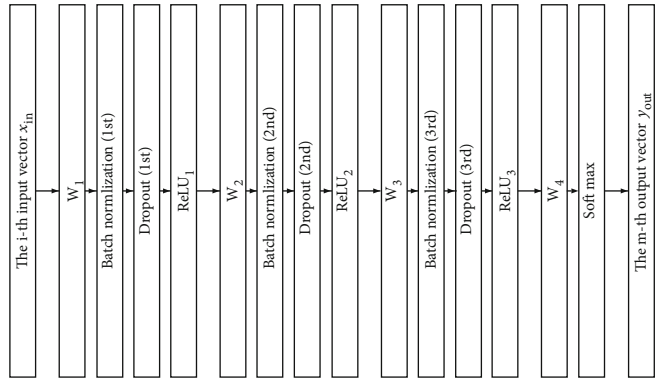


Figure 1: A block diagram of the proposed neural network structure to realize the forward signal propagation of machine learning.

consists of an input layer with N_{in} nodes (or neurons), an output layer with N_{out} nodes, and three hidden layers. Since $0 \le u_D^{(m)} \le N_{\text{data}}/10$, we set N_{out} to $N_{\text{data}}/10$ and let the output layer act as a multiclass classifier generating a one-hot encoded binary sequence of multiple zeros and single ones. The position of a single 1 in the one-hot encoded binary sequence represents $u_D^{(m)}$. The output signals of the three hidden layers are processed with the batch normalization function [39], the dropout function [40], and the rectified linear unit (ReLU) activation function [41]. The output signal of the output layer is processed with a softmax regression (SoftMax) activation function [42]. Since a bias factor of 1 is added to every sequence entering each hidden layer, the dimensions of the four weight matrices, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, and $\mathbf{W}_4$, are chosen as $(N_{\text{mid}} - 1) \times (N_{\text{in}} + 1)$, $(N_{\text{mid}} - 1) \times N_{\text{mid}}$, $(N_{\text{mid}} - 1) \times N_{\text{mid}}$, and $N_{\text{out}} \times N_{\text{mid}}$, respectively, where N_{mid} is selected as 1000. We randomly initialize all components of $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, and $\mathbf{W}_4$ using a Gaussian distribution with mean 0 and variance $\sqrt{2/N_{\text{col}}}$ as in [43], where N_{col} denotes the column number of the matrix to initialize. The CFR can be said to be a kind of raw CSI, and the averaged PSD sequence can be said to be a kind of processed CSI. In this paper, as two representative machine learning approaches, we consider a machine learning approach using the CFR sequence (i.e., raw CSI) as training data and a machine learning approach using an averaged PSD sequence (i.e., preprocessed CSI) as training data. We name the two machine learning approaches ML1 and ML2, respectively. In ML1, the training data are selected from a CFR sequences of length $N_F \times N_{\text{data}}$ (i.e., $N_{\text{in}} = N_F \times N_{\text{data}}$). The m-th training data for $m = 1, 2, \cdots, N_{\text{TUE}}$ can be written as

$$\mathbf{s}_m = \left[\mathbf{H}_F^{(m)}[0]^T \mathbf{H}_F^{(m)}[1]^T \cdots \mathbf{H}_F^{(m)}[N_{\text{data}} - 1]^T\right]^T, \tag{13}$$

where

$$\mathbf{H}_F^{(m)}[n] = \left[H_0^{(m)}[n] H_1^{(m)}[n] \cdots H_{N_F-1}^{(m)}[n]\right]^T. \tag{14}$$

In ML2, the training data are selected from the averaged PSD sequences of length $N_{\text{data}}/2$ (i.e., $N_{\text{in}} = N_{\text{data}}/2$). The m-th training data for $m = 1, 2, \cdots, N_{\text{TUE}}$ can be written as

$$\mathbf{s}_m = \left[s_{m,0} s_{m,1} \cdots s_{m,N_{\text{in}}-1}\right]^T, \tag{15}$$

where

$$s_{m,u} = \frac{1}{N_F} \sum_{k=0}^{N_F-1} S_k^{(m)}[u]. \tag{16}$$

The following is a summary of the notations used in machine learning algorithms:

(i) $\mathbf{x}(n)$ represents the n-th component of a vector or sequence, $\mathbf{x}$

(ii) $\mathbf{x}_1 e \mathbf{x}_2$ represents the component-wise product of the two vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$

(iii) $\mathbf{A}(k, :)$ represents the k-th row of the matrix $\mathbf{A}$

(iv) $\text{len}(\mathbf{x})$ represents the length of the $\mathbf{x}$ vector

(v) $\mathbf{x}(k : l)$ represents a column vector whose components are the components of the $\mathbf{x}$ vector from the k-th position to the l-th position

(vi) Given two vectors $\mathbf{x}$ and $\mathbf{y}$ of equal length, $(\mathbf{x} > \mathbf{y})$ represents a vector whose n-th component is 1 if $\mathbf{x}(n) > \mathbf{y}(n)$ and 0 if $\mathbf{x}(n) \le \mathbf{y}(n)$ for $n = 1, 2, \cdots, \text{len}(\mathbf{x})$

(vii) $\text{randn}(N_1, N_2)$ represents a randomly generated $N_1 \times N_2$ matrix whose components follow a Gaussian distribution with zero mean and unit variance

(viii) $(\cdot)^T$ represents the transpose operator

(ix) For input vector $\mathbf{x}$, $f_{\text{Sum}}(\mathbf{x})$ represents a function whose output is given as $\sum_{n=1}^{\text{len}(\mathbf{x})} \mathbf{x}(n)$

(x) $\text{diag}(\mathbf{x})$ represents a square matrix whose diagonal components are given by the components of the vector $\mathbf{x}$

1 Input:
2 **S** which denotes the data of the averaged PSD sequences
3 **T** which denotes the data of the target Doppler
4 shift index sequences
5
6 Fixed Parameters:
7 $\alpha_A = 0.001, \alpha_B = 0.01, R_{\mathrm{D}} = 0.02$
8
9 Weight Matrix Initialization:
10 $\mathbf{W}_1 = (2/\sqrt{N_{\mathrm{mid}} - 1}) \cdot \mathrm{randn}(N_{\mathrm{mid}} - 1, N_{\mathrm{in}} + 1)$
11 $\mathbf{W}_2 = (2/\sqrt{N_{\mathrm{mid}} - 1}) \cdot \mathrm{randn}(N_{\mathrm{mid}} - 1, N_{\mathrm{mid}})$
12 $\mathbf{W}_3 = (2/\sqrt{N_{\mathrm{mid}} - 1}) \cdot \mathrm{randn}(N_{\mathrm{mid}} - 1, N_{\mathrm{mid}})$
13 $\mathbf{W}_4 = (2/\sqrt{N_{\mathrm{out}}}) \cdot \mathrm{randn}(N_{\mathrm{out}}, N_{\mathrm{mid}})$
14 $\gamma_1 = 1, \beta_1 = 0, \gamma_2 = 1, \beta_2 = 0, \gamma_3 = 1, \beta_3 = 0$
15
16 for $i \in \{1, 2, \cdots, N_{\mathrm{epoch}}\}$ do
17 for $m \in \{1, 2, \cdots, N_{\mathrm{TUE}}\}$ do
18 $\mathbf{x}_{\mathrm{in}} = \mathbf{S}(m, :)^T$
19 $\mathbf{t} = \mathbf{T}(m, :)^T$
20 Forward Signal Propagation (Algorithm 2)
21 Backward Error Propagation (Algorithm 4)
22 Parameter Update (Algorithm 6)
23 end
24 end

Algorithm 1: Overall machine learning algorithm.

(xi) For two column vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, $[\mathbf{x}_1^T ; \mathbf{x}_2^T]$ represents a matrix created by stacking two row vectors $\mathbf{x}_1^T$ and $\mathbf{x}_2^T$

A full overview of the proposed machine learning algorithm is given in Algorithm 1. In both ML1 and ML2, the forward signal propagation algorithm, the backward error propagation algorithm, and the parameter update algorithm are performed $N_{\mathrm{epoch}} \times N_{\mathrm{TUE}}$ times in two "for" loops. In the forward signal propagation algorithm, the input vector $\mathbf{x}_{\mathrm{in}}$ is initialized with $\mathbf{s}_m$. The output vector $\mathbf{y}_{\mathrm{out}}$ resulting from $\mathbf{x}_{\mathrm{in}}$ is used to yield the final loss (or final cost) ξ. To compute ξ, the sequence of $\mathbf{d} = f_{\mathrm{SoftMax}}(\mathbf{t}_m)$ that satisfies $\sum_{n=1}^{N_{\mathrm{data}}} \mathbf{d}(n) = 1$ is prepared in advance, where $\mathbf{t}_m$ denotes the m-th target Doppler spread index sequence written as a one-hot encoded binary sequence consisting of multiple 0's and a single 1. One example of $\mathbf{t}_m$ can be written as

$$\mathbf{t}_m = [\,00 \cdots 010 \cdots 0\,]^T, \tag{17}$$

where the index of a single 1 in $\mathbf{t}_m$ represents the Doppler spread index defined in (11). The whole data that needs to be prepared in advance to run the proposed machine learning algorithm for Doppler spread estimation can be arranged in two matrices; first, the target Doppler spread index sequence matrix,

$$\mathbf{T} = \left[\mathbf{t}_1^T ; \mathbf{t}_2^T ; \cdots ; \mathbf{t}_{N_{\mathrm{TUE}}}^T\right], \tag{18}$$

and second, the training data matrix,

$$\mathbf{S} = \left[\mathbf{s}_1^T ; \mathbf{s}_2^T ; \cdots ; \mathbf{s}_{N_{\mathrm{TUE}}}^T\right]. \tag{19}$$

To gather information about **T**, the BS can have a TUE move with a constant velocity v and have that TUE report its Doppler spread index, $u_{\mathrm{D}} = T \cdot f_c \cdot v/c$, where $c = 3 \times 10^8$ (m/sec) and f_c is the carrier frequency. To gather information about **S**, the BS needs to estimate the CSI from that TUE and use the CSI to compute the training data. This method of collecting information by the BS entails system overhead. However, unless there are significant changes in Doppler spread statistics, reusing trained machine learning weights for Doppler spread estimation can reduce the overhead. Since it is difficult to actually collect the measured CSI at various times and places that can generate different Doppler spreads, in this paper, the collection of CSI necessary for machine learning or performance evaluation is limited to work through simulation.

By the double "for" loops of Algorithm 1, three subalgorithms, namely, "Forward Signal Propagation" (Algorithm 2), "Backward Error Propagation" (Algorithm 4), and "Parameter Update" (Algorithm 6) are executed iteratively. In Algorithm 2, the training data are propagated through the proposed neural network structure. Firstly, the input sequence $\mathbf{x}_{\mathrm{in}}$ is incremented by a bias factor 1 to produce the increased input sequence $\tilde{\mathbf{x}}_1$. Then, $\tilde{\mathbf{x}}_1$ is multiplied by $\mathbf{W}_1$ to produce $\check{\mathbf{x}}_1$. Then, $\check{\mathbf{x}}_1$ is batch-normalized to yield $\bar{\mathbf{x}}_1$. The batch normalization function [39] is written in Algorithm 3. Then, $\bar{\mathbf{x}}_1$ is activated by the ReLU function [41] to

$\tilde{\mathbf{x}}_1 = [1; \mathbf{x}_{\text{in}}]$
$\check{\mathbf{x}}_1 = \mathbf{W}_1 \cdot \tilde{\mathbf{x}}_1$
$\bar{\mathbf{x}}_1 = f_{\text{BatchNorm}}(\check{\mathbf{x}}_1, \gamma 1, \beta 1)$
$\hat{\mathbf{x}}_1 = f_{\text{ReLU}}(\bar{\mathbf{x}}_1)$
$\mathbf{x}_2 = f_{\text{Drop}}(\hat{\mathbf{x}}_1, R_{\text{D}})$
$\tilde{\mathbf{x}}_2 = [1; \mathbf{x}_2]$
$\check{\mathbf{x}}_2 = \mathbf{W}_2 \cdot \tilde{\mathbf{x}}_2$
$\bar{\mathbf{x}}_2 = f_{\text{BatchNorm}}(\check{\mathbf{x}}_2, \gamma_2, \beta_2)$
$\hat{\mathbf{x}}_2 = f_{\text{ReLU}}(\bar{\mathbf{x}}_2)$
$\mathbf{x}_3 = f_{\text{Drop}}(\hat{\mathbf{x}}_2, R_{\text{D}})$
$\tilde{\mathbf{x}}_3 = [1; \mathbf{x}_3]$
$\check{\mathbf{x}}_3 = \mathbf{W}_3 \cdot \tilde{\mathbf{x}}_3$
$\bar{\mathbf{x}}_3 = f_{\text{BatchNorm}}(\check{\mathbf{x}}_3, \gamma_3, \beta_3)$
$\hat{\mathbf{x}}_3 = f_{\text{ReLU}}(\bar{\mathbf{x}}_3)$
$\mathbf{x}_4 = f_{\text{Drop}}(\hat{\mathbf{x}}_3, R_{\text{D}})$
$\tilde{\mathbf{x}}_4 = [1; \mathbf{x}_4]$
$\check{\mathbf{x}}_4 = \mathbf{W}_4 \cdot \tilde{\mathbf{x}}_4$
$\mathbf{y}_{\text{out}} = f_{\text{SoftMax}}(\check{\mathbf{x}}_4)$

Algorithm 2: Forward signal propagation.

$\bar{\mathbf{x}} = f_{\text{BatchNorm}}(\check{\mathbf{x}}, \gamma, \beta)$
$M = \text{len}(\check{\mathbf{x}})$
$\mu = (1/M) f_{\text{Sum}}(\check{\mathbf{x}})$
$\sigma^2 = (1/M) f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot (\check{\mathbf{x}} - \mu))$
$\varepsilon = 10^{-7}$
$\mathbf{x} = \check{\mathbf{x}} - \mu / \sqrt{\sigma^2 + \varepsilon}$
$\bar{\mathbf{x}} = \gamma \cdot \mathbf{x} + \beta$
return $\bar{\mathbf{x}}$

Algorithm 3: Batch normalization function.

$\mathbf{d} = f_{\text{SoftMax}}(\mathbf{t})$
$\xi = -\sum_{n=1}^{N_{\text{data}}} \mathbf{d}(n) \cdot \log(\mathbf{y}_{\text{out}}(n))$
$\check{\mathbf{e}}_4 = \partial\xi/\partial\check{\mathbf{x}}_4 = \mathbf{y}_{\text{out}} - \mathbf{d}$
$\tilde{\mathbf{e}}_4 = \partial\xi/\partial\tilde{\mathbf{x}}_4 = \mathbf{W}_4^T \cdot \check{\mathbf{e}}_4$
$\hat{\mathbf{e}}_3 = \partial\xi/\partial\hat{\mathbf{x}}_3 = (\tilde{\mathbf{e}}_4(2 : \text{len}(\tilde{\mathbf{e}}_4)) \odot \mathbf{z}_{\text{D},3})/(1 - R_{\text{D}})$
$\bar{\mathbf{e}}_3 = \partial\xi/\partial\bar{\mathbf{x}}_3 = (\hat{\mathbf{x}}_3 > 0) \odot \hat{\mathbf{e}}_3$
$[\check{\mathbf{e}}_3, d\gamma_3, d\beta_3] = f_{\text{BackwardBatchNorm}}(\check{\mathbf{x}}_3, \gamma_3, \bar{\mathbf{e}}_3)$
$\tilde{\mathbf{e}}_3 = \partial\xi/\partial\tilde{\mathbf{x}}_3 = \mathbf{W}_3^T \cdot \check{\mathbf{e}}_3$
$\hat{\mathbf{e}}_2 = \partial\xi/\partial\hat{\mathbf{x}}_2 = (\tilde{\mathbf{e}}_3(2 : \text{len}(\tilde{\mathbf{e}}_3)) \odot \mathbf{z}_{\text{D},2})/(1 - R_{\text{D}})$
$\bar{\mathbf{e}}_2 = \partial\xi/\partial\bar{\mathbf{x}}_2 = (\hat{\mathbf{x}}_2 > 0) \odot \hat{\mathbf{e}}_2$
$[\check{\mathbf{e}}_2, d\gamma_2, d\beta_2] = f_{\text{BackwardBatchNorm}}(\check{\mathbf{x}}_2, \gamma_2, \bar{\mathbf{e}}_2)$
$\tilde{\mathbf{e}}_2 = \partial\xi/\partial\tilde{\mathbf{x}}_2 = \mathbf{W}_2^T \cdot \check{\mathbf{e}}_2$
$\hat{\mathbf{e}}_1 = \partial\xi/\partial\hat{\mathbf{x}}_1 = (\tilde{\mathbf{e}}_2(2 : \text{len}(\tilde{\mathbf{e}}_2)) \odot \mathbf{z}_{\text{D},1})/(1 - R_{\text{D}})$
$\bar{\mathbf{e}}_1 = \partial\xi/\partial\bar{\mathbf{x}}_1 = (\hat{\mathbf{x}}_1 > 0) \odot \hat{\mathbf{e}}_1$
$[\check{\mathbf{e}}_1, d\gamma_1, d\beta_1] = f_{\text{BackwardBatchNorm}}(\check{\mathbf{x}}_1, \gamma_1, \bar{\mathbf{e}}_1)$

Algorithm 4: Backward error propagation.

generate $\hat{\mathbf{x}}_1$. The ReLU function can be written as

$$f_{\text{ReLU}}(\bar{\mathbf{x}}) = [f_{\text{ReLU}}(\bar{\mathbf{x}}(1)) \cdots f_{\text{ReLU}}(\bar{\mathbf{x}}(\text{len}(\bar{\mathbf{x}})))]^T, \quad (20)$$

where

$$f_{\text{ReLU}}(\bar{x}) = \begin{cases} \bar{x} & \text{if } \bar{x} > 0, \\ 0 & \text{otherwise} \end{cases}. \quad (21)$$

Then, $\hat{\mathbf{x}}_1$ is applied to the dropout function [40] resulting in $\mathbf{x}_2$. Dropout is a technique to solve the overfitting problem by omitting hidden nodes with predefined probabilities. The dropout function can be written as

$$\mathbf{x}_2 = f_{\text{Drop}}(\hat{\mathbf{x}}_1, R_D) = \hat{\mathbf{x}}_1 \odot \left(\frac{1}{1 - R_D} \cdot \mathbf{z}_{D,1}\right), \quad (22)$$

where R_D denotes the dropout rate given by 0.02. $\mathbf{z}_{D,i}$ for $i = 1, 2, 3$ denotes a randomly generated vector of zeros and ones that satisfy the condition that the number of zeros is equal to the product of the value of R_D and the number of components in $\hat{\mathbf{x}}_1$. In (22), for the purpose of maintaining the same power of $\mathbf{x}_2$ as in $\hat{\mathbf{x}}_1$, $\mathbf{z}_{R_D,1}$ is multiplied by $1/(R_D - 1)$. Then, $\tilde{\mathbf{x}}_2$ is obtained by inserting a bias factor of 1 into $\mathbf{x}_2$. Similar to the process above, $\tilde{\mathbf{x}}_3$ and $\tilde{\mathbf{x}}_4$ will also be obtained. Then, $\mathbf{W}_4$ is multiplied by $\tilde{\mathbf{x}}_4$ to produce $\check{\mathbf{x}}_4$. Finally, $\mathbf{y}_{\text{out}}$ is created by activating $\check{\mathbf{x}}_4$ with SoftMax function ([42]), i.e., $\mathbf{y}_{\text{out}} = f_{\text{SoftMax}}(\check{\mathbf{x}}_4)$, where

$$\mathbf{y}_{\text{out}}(n) = \frac{e^{\check{\mathbf{x}}_4(n)}}{\sum_{i=1}^{N_{\text{out}}} e^{\check{\mathbf{x}}_4(i)}}, \quad (23)$$

for $n = 1, \cdots, N_{\text{out}}$. In Algorithm 4, the back-propagation error is derived using the backward error propagation technique [44, 45]. To determine the final cost (or final loss) ξ, cross-entropy function [42] is used with two input vectors $\mathbf{y}_{\text{out}}$ and $\mathbf{d}$ as

$$\xi = -\sum_{n=1}^{N_{\text{data}}} \mathbf{d}(n) \cdot \log(\mathbf{y}_{\text{out}}(n)). \quad (24)$$

When an error of amplitude 1 is back-propagated from the output position of the cross-entropy function to the n-th input position of the SoftMax function, a back-propagation error is induced as

$$\frac{\partial\xi}{\partial\check{\mathbf{x}}_4(n)} = \mathbf{y}_{\text{out}}(n) - \mathbf{d}(n). \quad (25)$$

Therefore, if an error of amplitude 1 propagates back from the output position of the cross-entropy function to the input vector position of the SoftMax function, the resulting back-propagation error can be written as

$$\check{\mathbf{e}}_4 = \frac{\partial\xi}{\partial\check{\mathbf{x}}_4} = \mathbf{y}_{\text{out}} - \mathbf{d}. \quad (26)$$

$M = \text{len}(\check{\mathbf{x}})$
$\mu = (1/M) f_{\text{Sum}}(\check{\mathbf{x}})$
$\sigma^2 = (1/M) f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot (\check{\mathbf{x}} - \mu))$
$\varepsilon = 10^{-7}$
$\mathbf{5x} = \check{\mathbf{x}} - \mu / \sqrt{\sigma^2 + \varepsilon}$
$d\beta = f_{\text{Sum}}(\bar{\mathbf{e}})$
$d\gamma = f_{\text{Sum}}(\mathbf{x} \odot \bar{\mathbf{e}})$
$\check{\mathbf{e}} = (\gamma / M\sqrt{\sigma^2 + \varepsilon})(M \cdot \bar{\mathbf{e}} - f_{\text{Sum}}(\bar{\mathbf{e}}) \cdot 1_M - (f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}) / \sigma^2 + \varepsilon)(\check{\mathbf{x}} - \mu))$
return $[\check{\mathbf{e}}, d\gamma, d\beta]$

Algorithm 5: Backward error propagation of batch normalization function $[\check{\mathbf{e}}, d\gamma, d\beta] = f_{\text{BackwardBatchNorm}}(\check{\mathbf{x}}, \gamma, \bar{\mathbf{e}})$.

$\gamma_3 = \gamma_3 - \alpha_A \cdot d\gamma_3$
$\beta_3 = \beta_3 - \alpha_A \cdot d\beta_3$
$\gamma_2 = \gamma_2 - \alpha_A \cdot d\gamma_2$
$\beta_2 = \beta_2 - \alpha_A \cdot d\beta_2$
$\gamma_1 = \gamma_1 - \alpha_A \cdot d\gamma_1$
$\beta_1 = \beta_1 - \alpha_A \cdot d\beta_1$
$\mathbf{W}_4 = \mathbf{W}_4 - \alpha_B \cdot \check{\mathbf{e}}_4 \cdot \tilde{\mathbf{x}}_4^T$
$\mathbf{W}_3 = \mathbf{W}_3 - \alpha_B \cdot \check{\mathbf{e}}_3 \cdot \tilde{\mathbf{x}}_3^T$
$\mathbf{W}_2 = \mathbf{W}_2 - \alpha_B \cdot \check{\mathbf{e}}_2 \cdot \tilde{\mathbf{x}}_2^T$
$\mathbf{W}_1 = \mathbf{W}_1 - \alpha_B \cdot \check{\mathbf{e}}_1 \cdot \tilde{\mathbf{x}}_1^T$

Algorithm 6: Parameter update.

Since $\check{\mathbf{x}}_4 = \mathbf{W}_4 \cdot \tilde{\mathbf{x}}_4$, we derive

$$\frac{d\check{\mathbf{x}}_4}{d\tilde{\mathbf{x}}_4} = \mathbf{W}_4^T. \tag{27}$$

By the chain rule [46] accompanying (26) and (27), back-propagation error generated at the input position of $\mathbf{W}_4$ can be written as

$$\tilde{\mathbf{e}}_4 = \frac{\partial \xi}{\partial \tilde{\mathbf{x}}_4} = \frac{d\check{\mathbf{x}}_4}{d\tilde{\mathbf{x}}_4} \cdot \frac{\partial \xi}{\partial \check{\mathbf{x}}_4} = \mathbf{W}^T \cdot \check{\mathbf{e}}_4. \tag{28}$$

Taking into account the bias factor contained in $\tilde{\mathbf{x}}_4$, we derive

$$\frac{d\tilde{\mathbf{x}}_4}{d\hat{\mathbf{x}}_3} = \left[0_{N_{\text{mid}}-1} \frac{1}{1 - R_D} \cdot \text{diag}\,(\mathbf{z}_{D,3}) \right], \tag{29}$$

where $0_{N_{\text{mid}}-1}$ denotes a zero-column vector of length $N_{\text{mid}} - 1$. Therefore, the back-propagation error at the input position of the dropout function can be written as

$$\hat{\mathbf{e}}_3 = \frac{\partial \xi}{\partial \hat{\mathbf{x}}_3} = \frac{\partial \tilde{\mathbf{x}}_4}{\partial \hat{\mathbf{x}}_3} \cdot \frac{\partial \xi}{\partial \tilde{\mathbf{x}}_4} = \frac{\tilde{\mathbf{e}}_4(2 : \text{len}(\tilde{\mathbf{e}}_4)) \odot \mathbf{z}_{D,3}}{1 - R_D}. \tag{30}$$

Since $\hat{\mathbf{x}}_3 = f_{\text{ReLU}}(\bar{\mathbf{x}}_3)$, we can write

$$\frac{\partial \hat{\mathbf{x}}_3(n)}{\partial \bar{\mathbf{x}}_3(m)} = \begin{cases} (\hat{\mathbf{x}}_3(n) > 0) & \text{if } n = m, \\ 0 & \text{otherwise,} \end{cases} \tag{31}$$

for $n, m = 1, 2, \cdots, N_{\text{data}} - 1$. By expanding (31), we derive

$$\frac{\partial \hat{\mathbf{x}}_3}{\partial \bar{\mathbf{x}}_3} = \text{diag}\,((\hat{\mathbf{x}}_3 > 0)). \tag{32}$$

From the result of (30) and (32), the back-propagation error at the input position of the ReLU function can be written as

$$\bar{\mathbf{e}}_3 = \frac{\partial \xi}{\partial \bar{\mathbf{x}}_3} = \frac{\partial \hat{\mathbf{x}}_3}{\partial \bar{\mathbf{x}}_3} \cdot \frac{\partial \xi}{\partial \hat{\mathbf{x}}_3} = (\hat{\mathbf{x}}_3 > 0) \odot \hat{\mathbf{e}}_3. \tag{33}$$

The backward error propagation function for batch normalization is defined in Algorithm 5, whose detailed derivation is provided in the Appendix. Applying $\bar{\mathbf{e}}_3$ to that function will result in the back-propagation error, $\check{\mathbf{e}}_3$, at the input vector position of the batch normalization function. Similar to the process above, $\check{\mathbf{e}}_2$ and $\check{\mathbf{e}}_1$ can also be obtained. In Algorithm 6, batch normalization parameters, γ_1, β_1, γ_2, β_2, γ_3, and β_3, and weight matrices, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, and $\mathbf{W}_4$, are updated according to stochastic gradient descent [47]. The batch normalization parameters, γ_i and β_i for $i = 1, 2, 3$, are updated according to

$$\begin{aligned} \gamma_i &= \gamma_i - \alpha_A \cdot d\gamma_i, \\ \beta_i &= \beta_i - \alpha_A \cdot d\beta_i, \end{aligned} \tag{34}$$

where α_A denotes the learning rate chosen as 0.001. The values of $d\gamma_i$ and $d\beta_i$ are generated by the backward batch normalization function, the detailed derivation of which is presented in the Appendix. $\mathbf{W}_4$ is updated according to the stochastic steepest descent method as

$$\mathbf{W}_4 = \mathbf{W}_4 - \alpha_B \cdot \frac{\partial \xi}{\partial \mathbf{W}_4}, \tag{35}$$

where α_B denotes the learning rate chosen as 0.01. To find

$\partial\xi/\partial\mathbf{W}_4$, we first derive

$$\frac{\partial\xi}{\partial\mathbf{W}_4(i,j)} = \frac{\partial\check{\mathbf{x}}_4}{\partial\mathbf{W}_4(i,j)} \cdot \frac{\partial\xi}{\partial\check{\mathbf{x}}_4} = \frac{\partial\check{\mathbf{x}}_4}{\partial\mathbf{W}_4(i,j)} \cdot \check{\mathbf{e}}_4. \tag{36}$$

Since $\check{\mathbf{x}}_4 = \mathbf{W}_4 \cdot \tilde{\mathbf{x}}_4$, we derive

$$\frac{\partial\check{\mathbf{x}}_4}{\partial\mathbf{W}_4(i,j)} = \tilde{\mathbf{x}}_4(j) \cdot \mathbf{u}_i^T. \tag{37}$$

$\mathbf{u}_i$ denotes a vector of length len($\tilde{\mathbf{x}}_4$), where the i-th component is given as 1 and the other components are given as 0's. By substituting the result of (37) into (36), we derive

$$\frac{\partial\xi}{\partial\mathbf{W}_4(i,j)} = \tilde{\mathbf{x}}_4(j) \cdot \mathbf{u}_i^T \cdot \check{\mathbf{e}}_4 = \check{\mathbf{e}}_4(i) \cdot \tilde{\mathbf{x}}_4(j). \tag{38}$$

By stacking the results of (38) for all indices of i and j, we can write

$$\frac{\partial\xi}{\partial\mathbf{W}_4} = \check{\mathbf{e}}_4 \cdot \tilde{\mathbf{x}}_4^T. \tag{39}$$

By substituting the result of (39) into (35), we derive

$$\mathbf{W}_4 = \mathbf{W}_4 - \alpha_B \cdot \check{\mathbf{e}}_4 \cdot \tilde{\mathbf{x}}_4^T. \tag{40}$$

Similar to the process above, the formula for updating $\mathbf{W}_i$ for $i = 1, 2, 3$ can also be derived as

$$\mathbf{W}_i = \mathbf{W}_i - \alpha_B \cdot \check{\mathbf{e}}_i \cdot \tilde{\mathbf{x}}_i^T. \tag{41}$$

Note that the BS can train the weight matrices offline using Algorithm 1. The trained weight matrix can be used by the BS to estimate the Doppler spread of the target user equipment (UE) online. When the weight matrices are trained by ML1, the BS can find the Doppler spread of the target UE by computing the CFR sequence based on the CSI of the target UE and using the CFR sequence as $\mathbf{x}_{\text{in}}$ in Algorithm 2 to generate $\check{\mathbf{x}}_4$. When the weight matrices are trained by ML2, the BS can find the Doppler spread of the target UE by calculating the averaged PSD sequence based on the CSI of the target UE for $\mathbf{x}_{\text{in}}$ and applying $\mathbf{x}_{\text{in}}$ to Algorithm 2 to find $\check{\mathbf{x}}_4$. Since the maximum component index of $\check{\mathbf{x}}_4$ represents the Doppler spread index u_{D}, the Doppler spread of the target UE is given by $f_{\text{D}} = u_{\text{D}}/T$ based on (11).

4. Simulation Results

In the simulation, OFDM system parameters are chosen as $N_F = 128$, $L = 5$, and $T_S = 0.001$ sec. The number of pilot subcarriers is chosen to be $N_P = 8$, which means that the subcarrier spacing between two neighboring pilots is given by $N_F/N_P = 16$. The machine learning settings for two machine learning approaches, ML1 and ML2, are summarized in Table 1.

To evaluate the Doppler spread estimation performance, we consider the root mean square error (RMSE)

$$\text{RMSE} = \left(\frac{1}{N_M}\sum_{m=1}^{N_M}\left(f\wedge_D^{(m)} - f_D^{(m)}\right)^2\right)^{1/2}, \tag{42}$$

and the normalized RMSE (NRMSE)

$$\text{NRMSE} = \left(\frac{1}{N_M}\sum_{m=1}^{N_M}\left(\frac{f\wedge_D^{(m)} - f_D^{(m)}}{f_D^{(m)}}\right)^2\right)^{1/2}, \tag{43}$$

where $\widehat{f}_{\text{D}}^{(m)}$ denotes the estimated Doppler spread at the m-th TUE and N_{M} denotes the Monte Carlo simulation number, which is set to 10^5. For comparison, the two previous Doppler estimation approaches introduced in [5, 6] are also considered and named PREV1 and PREV2, respectively. PREV1 estimates Doppler spread by referring to the maximum position of the maxima of the multiple PSD functions derived from channel multipaths [5]. PREV2 estimates Doppler spread by referencing the maximum position of the averaged PSD values greater than 10% of the maximum averaged PSD value [6].

In Figure 2, we evaluated the RMSEs of ML1, ML2, PREV1, and PREV2 in terms of SNR in a Rayleigh and isotropic channel scenario, where f_D, K_r, κ, and α were assumed to be 40 Hz, 0, 0, and 0°, respectively. The same channel scenario was used to generate the CSI used for training ML1 and ML2 in the simulation. In the figure, the curve for ML2 is not shown because all RMSEs for ML2 are given as zero. This means that ML2 perfectly estimated the Doppler spread at all SNRs, unlike other approaches. PREV2 outperformed PREV1 and ML1 at SNRs above -0.8 dB, while PREV2 degraded as SNR decreased, performing worse than PREV1 and ML1 at SNRs below -2.5 dB. The reason for the poor performance of PREV2 at low SNR is that the maximum position of the averaged PSD values greater than 10% of the maximum averaged PSD value becomes more uncertain due to the PSD values adversely affected by small SNRs or large channel estimation errors. Although ML1 had higher implementation complexity than ML2, it performed much worse than ML2 because it could not effectively train the weight matrices using 2000(=N_{TUE}) CFR sequences. This indicates the effectiveness of using averaged PSD sequences as training data in machine learning for Doppler spread estimation compared to using CFR sequences when the number of the training data is limited.

Depending on the location and velocity of the UE and the scattering environment between the UE and the BS, channel characteristic variables such as SNR, f_D, θ, K_r, and α can be randomly generated. For the purpose of making ML1 and ML2 work well anytime and anywhere, in the following simulations, ML1 and ML2 were trained based on the information of the channels generated assuming a mixed-channel scenario in which SNR, f_D, θ, K_r, and $\alpha(=\alpha_0)$ have a uniform distribution between -6 dB and 30 dB, between

Table 1

Machine learning settings			
Parameter	Value	Parameter	Value
N_{TUE} (training sequence #)	2000	$N_{\text{data}} = T/T_s$	1000
N_{in} (input node #) in ML1	128000	N_{out} (output node #) in ML1	100
N_{in} (input node #) in ML2	500	N_{out} (output node #) in ML2	100
Hidden and output layer #	4	Epoch #	2000
Neuron # in $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$	1000	Neuron # in $\mathbf{W}_4$	100
Initial values of weights	Random	Dropout ratio	2%
Activation function	SoftMax (23)	Cost function	Cross entropy (24)

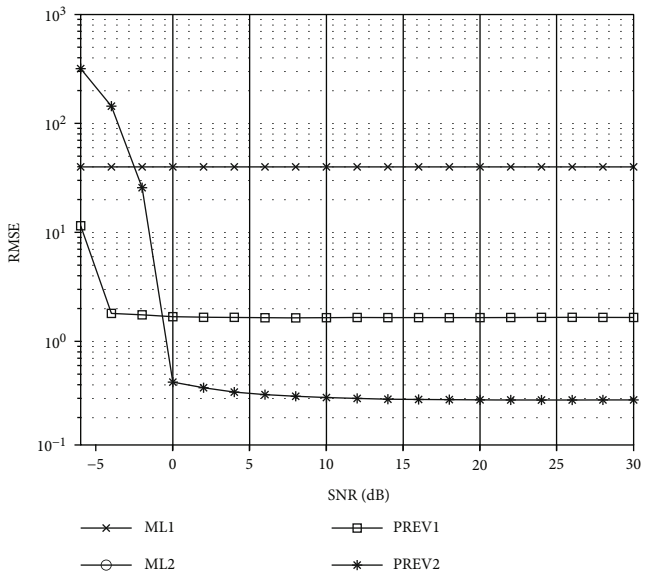


Figure 2: Comparison of RMSE in terms of SNR when f_D was 40 Hz, K_r was 0, κ was 0, α_0 was 0°, and α was 0°.

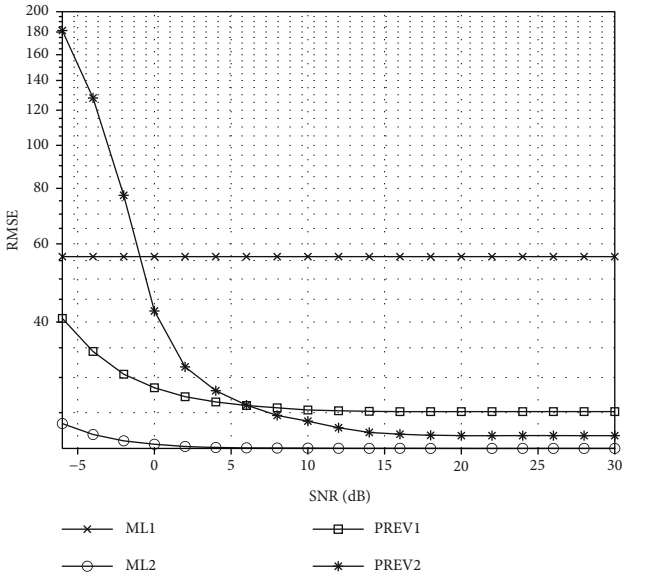


Figure 3: Comparison of RMSE in terms of SNR when f_D, K_r, θ, and $\alpha(=\alpha_0)$ have a uniform distribution between 0 Hz and 100 Hz, between 0 and 2, between 10° and 30°, and between -180° and 180°, respectively.

0 Hz and 100 Hz, between 10° and 30°, between 0 and 2, and between -180° and 180°, respectively.

In Figure 3, we evaluated the RMSEs of ML1, ML2, PREV1, and PREV2 in terms of SNR for the channels whose characteristic variables, f_D, K_r, θ, and α, have a uniform distribution between 0 Hz and 100 Hz, between 0 and 2, between 10° and 30°, and between -180° and 180°, respectively. α_0 was chosen equal to α as in [36]. It was observed that ML2 outperformed the other approaches at all SNRs because ML2 effectively trained the weight matrices in the mixed-channel scenario. ML2 yielded RMSE values of less than 21 for SNR above 0 dB, which means that when the carrier frequency is 2.4 GHz and the SNR is above 0 dB, the amount of error in user speed estimation is less than 9.45 km/hr. On the contrary, ML1 performed worst for SNRs higher than -1 dB because the 2000 CFR sequences were not sufficient to properly train ML1's weight matrices. Since ML1 used unprocessed CSI (i.e., CFR sequences) for Doppler spread estimation, it yielded a constant RMSE curve. PREV2 outperformed PREV1 at SNRs above 6 dB, while PREV2 degraded as SNR decreased, performing worse than PREV1 at SNRs below 6 dB. The reason for the poor performance of PREV2 at low SNR is that the maximum position of the averaged PSD values greater than 10% of the maximum averaged PSD value becomes more uncertain due to the PSD values adversely affected by small SNRs or large channel estimation errors.

In Figure 4, we evaluated the NRMSEs of ML1, ML2, PREV1, and PREV2 in terms of f_D for the channels whose characteristic variables, SNR, K_r, θ, and $\alpha(=\alpha_0)$, have a uniform distribution between -6 dB and 30 dB, between 0 and 2, between 10° and 30°, and between -180° and 180°, respectively. It was observed that ML2 outperformed the other approaches for all values of f_D because ML2 effectively trained the weight matrices in the mixed-channel scenario. ML1 yielded a constant NRMSE curve because it used unprocessed CSI (i.e., CFR sequences) for Doppler spread estimation. ML1 performed worst when f_D was greater than 53 Hz. PREV2 performed worse than PREV1 at all f_D values because PREV2 tended to yield very large NRMSEs at low SNRs, which led the overall performance of PREV2 to be

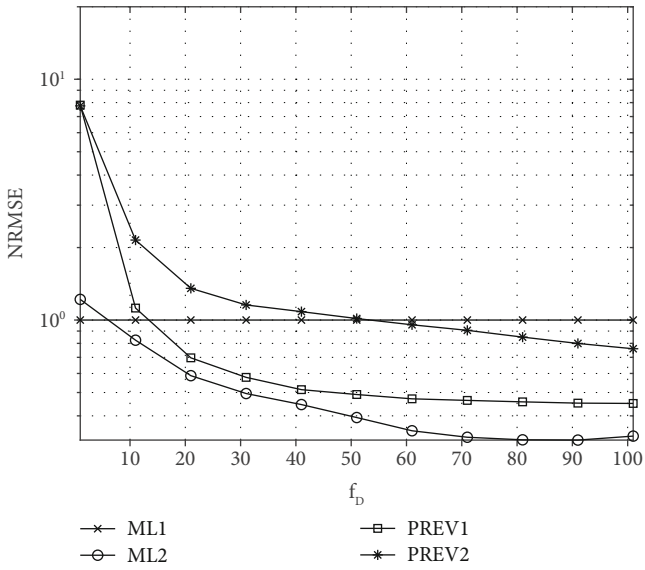


FIGURE 4: Comparison of RMSE in terms of f_D when SNR, K_r, θ, and $\alpha(=\alpha_0)$ have a uniform distribution between -6 dB and 30 dB, between 0 and 2, between 10° and 30°, and between -180° and 180°, respectively.

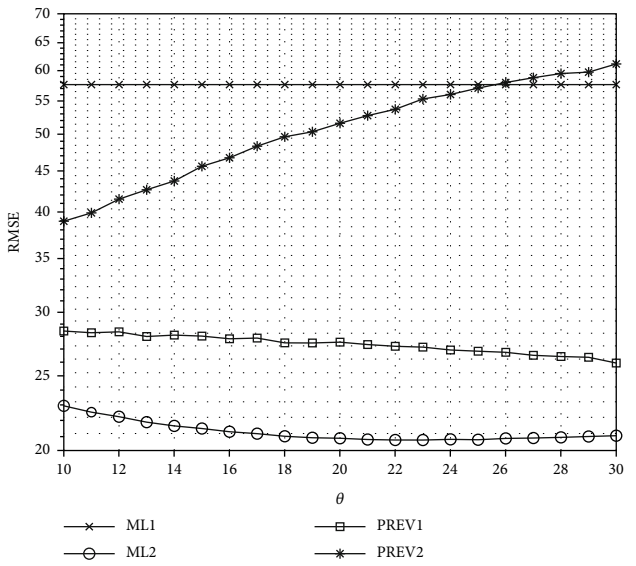


FIGURE 5: Comparison of RMSE in terms of θ when SNR, f_D, K_r, and $\alpha(=\alpha_0)$ have a uniform distribution between -6 dB and 30 dB, between 0 Hz and 100 Hz, between 0 and 2, and between -180° and 180°, respectively.

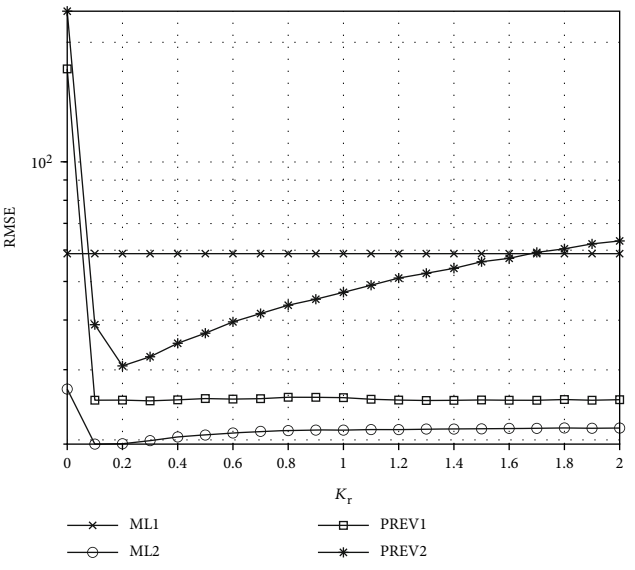


FIGURE 6: Comparison of RMSE in terms of K_r when SNR, f_D, θ, and $\alpha(=\alpha_0)$ have a uniform distribution between -6 dB and 30 dB, between 0 Hz and 100 Hz, between 10° and 30°, and between -180° and 180°, respectively.

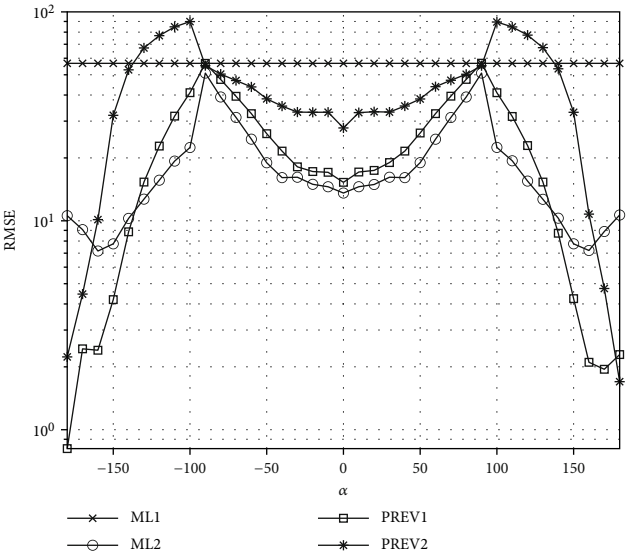


FIGURE 7: Comparison of RMSE in terms of $\alpha(=\alpha_0)$ when SNR, f_D, θ, and K_r have a uniform distribution between -6 dB and 30 dB, between 0 Hz and 100 Hz, between 10° and 30°, and between 0 and 2, respectively.

worse than PREV1 when the SNR was uniformly generated between -6 dB and 30 dB.

In Figure 5, we evaluated the RMSEs of ML1, ML2, PREV1, and PREV2 in terms of θ for the channels whose characteristic variables, SNR, f_D, K_r, and $\alpha(=\alpha_0)$, have a uniform distribution between −6 dB and 30 dB, between 0 Hz and 100 Hz, between 0 and 2, and between -180° and 180°, respectively. Note that κ representing the azimuth AOA width can be expressed as θ through $\kappa = (360°/\theta/\pi)^2$ [37]. It was observed that ML2 outperformed the other approaches at all values of θ because ML2 effectively trained the weight matrices in the mixed-channel scenario. PREV1 outperformed ML1 and PREV2 for all θ values. As θ increased, the performance of PREV2 decreased, which can be explained as follows. As θ increases, the channel autocorrelation takes a sharper shape according to (5). With sharper shaped channel autocorrelation, the PSD function forms a smoother shape according to (10), resulting in a smaller maximum PSD value. Therefore, the maximum position of the averaged PSD values greater than 10% of the maximum averaged PSD value becomes more uncertain by a larger θ,

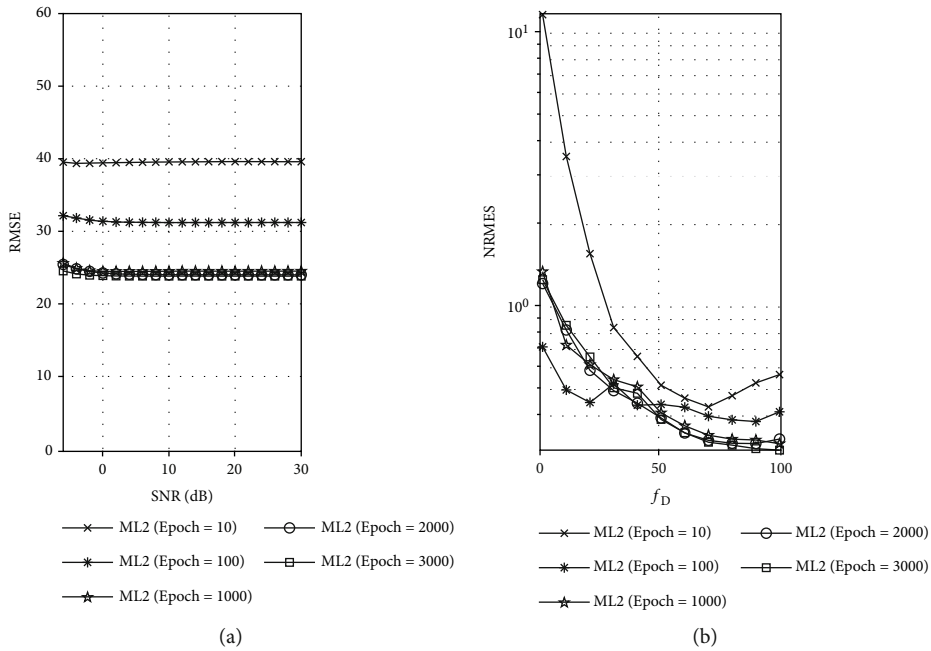


FIGURE 8: (a) Comparison of RMSE of ML2 in terms of SNR for various values of N_{TUE} by using the same simulation parameters as given in Figure 3. (b) Comparison of NRMSE of ML2 in terms of f_D for various values of N_{TUE} by using the same simulation parameters as given in Figure 4.

especially if the PSD values are adversely affected by a small SNR or large channel estimation error.

In Figure 6, we evaluated the RMSEs of ML1, ML2, PREV1, and PREV2 in terms of Rician K-factor K_r for the channels whose characteristic variables, SNR, f_D, θ, and $\alpha(=\alpha_0)$, have a uniform distribution between -6 dB and 30 dB, between 0 Hz and 100 Hz, between 10° and 30°, and between -180° and 180°, respectively. It was observed that ML2 outperformed the other approaches at all K_r values because ML2 effectively trained the weight matrices in the mixed-channel scenario. PREV2 degraded as K_r increased, which can be explained as follows. As K_r increases, the effect of the fixed channel component becomes more significant than the fading (or fluctuating) channel component. The greater the effect of the fixed channel component, the more imprecise it is to find the Doppler spread by referencing the maximum position of the averaged PSD values greater than 10% of the maximum averaged PSD value. Therefore, the performance of PREV2 degrades as K_r increases. At high SNR, PREV1, which estimates the Doppler spread by referring to the maximum position of the maxima of the multiple PSD functions derived from channel multipaths, outperformed ML1 and PREV2 because at least one channel multipath can have a strong fading component with high probability [5] and PREV1 could benefit from that multipath. Very small values of K_r cause very large RMSEs in PREV1 and PREV1. This is because for very small K_r values, a low SNR or a large channel estimation error greatly deteriorates their Doppler spread estimation performance and thus causes a large overall Doppler spread estimation error even if the SNR is uniformly generated.

In Figure 7, we evaluated the RMSEs of ML1, ML2, PREV1, and PREV2 in terms of mean direction of the azimuth AOA α for the channels whose characteristic variables, SNR, f_D, θ, K_r, have a uniform distribution between -6 dB and 30 dB, between 0 Hz and 100 Hz, between 10° and 30°, and between 0 and 2, respectively. α_0 was chosen equal to α as in [36]. It was observed that ML2 outperformed the other approaches for most α values. However, for α values with $165° < |\alpha| < 180°$, ML2 performed worse than PREV1 and PREV2. The reason for this phenomenon is that ML2 trained the weight matrices with the aim of improving the overall Doppler spread estimation performance for all α values while effectively avoiding the overfitting problem. It was observed that PREV1, PREV2, and ML2 produced RMSE curves that fluctuate in terms of α because PREV1, PREV2, and ML2 used PSD sequences for Doppler spread estimation. Notice the fact that the maximum position of the PSD function fluctuates due to the cos α_0 term as shown in (6). However, ML1 yielded a constant RMSE curve because it used unprocessed CSI (i.e., CFR sequences) for Doppler spread estimation. Because PREV1 and PREV2 assumed $\alpha_0 = 0°$ and estimated the Doppler spread, the performance of PREV1 and PREV2 deteriorated as $\alpha_0(=\alpha)$ value changed from 0° to 90°.

In Figures 8(a) and 8(b), we investigated the impact of N_{TUE} on the performance of ML2. In Figure 8(a), we evaluated the RMSE in terms of SNR using the same simulation parameters as given in Figure 3. In Figure 8(b), we evaluated the NRMSE in terms of f_D using the same simulation parameters as given in Figure 4. It can be seen from Figures 8(a) and 8(b) that the RMSE and NRMSE performance improved with increasing N_{TUE}. However, increasing

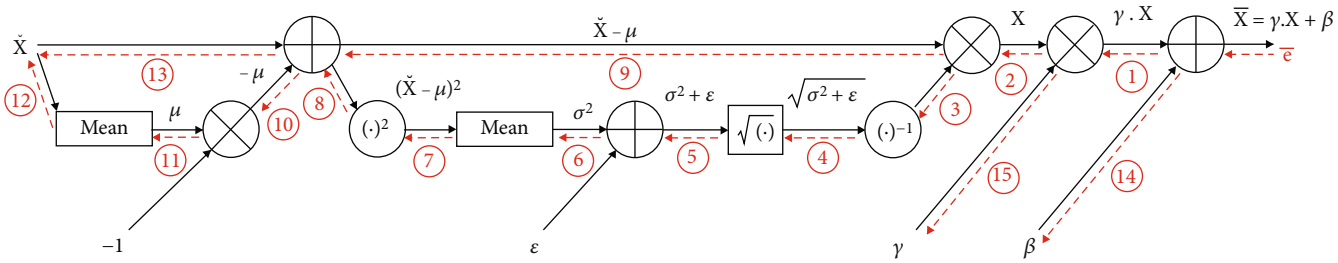


FIGURE 9: Computational graph of the back-propagation errors at all intermediate edges of the batch normalization function given in Algorithm 5 [48].

N_{TUE} to a value greater than 2000 produced only minor gains. From the above simulation results, we conclude that 2000 is a good choice for the N_{TUE} value in ML2.

5. Conclusion

A neural network structure in machine learning for Doppler spread estimation was proposed for an OFDM system. The weight matrix update algorithm was derived with the help of the backward error propagation technique and the stochastic steepest descent method. Numerical simulations have shown that averaged PSD sequences can be used to effectively train machine learning weights for Doppler spread estimation. The proposed machine learning approach using averaged PSD sequence as training data outperformed other Doppler spread estimation approaches under various channel conditions.

Appendix

A. The Derivation of the Back-Propagation Errors at All Edges of the Batch Normalization Function

The backward error propagation of the batch normalization function was derived in [48]. We present the back-propagation error at every edge of the batch normalization function with respect to the parameters defined in this paper. First, we summarize the backward error propagation properties [48] for some basic functions as follows:

(i) Given two vector inputs, $\mathbf{x}$ and $\mathbf{a}$, the back-propagation error through the "component wise addition" function arriving at the position of $\mathbf{x}$ is the same as the error given at the output position of that function. This property is used to derive the results between ① and ⑬ in Figure 9

(ii) Given two scalar inputs, x and a, the back-propagation error through the "addition" function arriving at the position of x is the same as the error given at the output position of that function. This property is used to derive the results between ⑥, ⑩, ⑭, and ⑮ in Figure 9

(iii) Given two vector inputs, $\mathbf{x}$ and $\mathbf{a}$, the back-propagation error through the "component wise multiplication" function arriving at the position of $\mathbf{x}$ is given by the component wise multiplication of $\mathbf{a}$ and the error given at the output position of that function. This property is used to derive the results between ①, ②, and ⑨ in Figure 9

(iv) Given two scalar inputs, x and a, the back-propagation error through the "multiplication" function arriving at the position of x is given by the produce of a and the error given at the output position of that function. This property is used to derive the results between ③ and ⑪ in Figure 9

(v) Given a vector input of length M, the back-propagation error through the "mean" function arriving at the input position of that function is given by the product of $(1/M)1_M$ and the error given at the output position of that function, where 1_M denotes a vector of length M consisting of all 1 s. This property is used to derive the results between ⑦ and ⑫ in Figure 9

(vi) Given a scalar input x, the back-propagation error through the "reciprocal" function arriving at the position of x is given by the product of $-x^{-2}$ and the error given at the output position of that function. This property is used to derive the results from ④ in Figure 9

(vii) Given a scalar input x, the back-propagation error through the "square root" function arriving at the position of x is given by the product of $(1/2)x^{-1/2}$ and the error given at the output position of that function. This property is used to derive the results from ⑤ in Figure 9

(viii) Given a vector input $\mathbf{x}$, the back-propagation error through the "componentwise-square" function arriving at the position of $\mathbf{x}$ is given by the product of $2\mathbf{x}$ and the error vector given at the output position of that function. This property is used to derive the results from ⑧ in Figure 9

Second, we summarize the back-propagation errors at all edges denoted by circled numbers as shown in Figure 9, which were derived based on the properties

mentioned above.

$$
\begin{aligned}
&① : \bar{\mathbf{e}},\\
&② : \gamma \cdot \bar{\mathbf{e}},\\
&③ : \gamma \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}),\\
&④ : -\frac{\gamma}{\sigma^2 + \varepsilon} \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}),\\
&⑤ : -\frac{\gamma}{2}\left(\sigma^2 + \varepsilon\right)^{-3/2} \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}),\\
&⑥ : -\frac{\gamma}{2}\left(\sigma^2 + \varepsilon\right)^{-3/2} \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}),\\
&⑦ : -\frac{\gamma}{2M}\left(\sigma^2 + \varepsilon\right)^{-3/2} \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}) \cdot 1_M,\\
&⑧ : -\frac{\gamma}{M}\left(\sigma^2 + \varepsilon\right)^{-3/2} \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}) \cdot (\check{\mathbf{x}} - \mu),\\
&⑨ : \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \cdot \bar{\mathbf{e}},\\
&⑩ : \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \cdot f_{\text{Sum}}(\bar{\mathbf{e}})\\
&-\frac{\gamma}{M}\left(\sigma^2 + \varepsilon\right)^{-3/2} \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}) \cdot f_{\text{Sum}}(\check{\mathbf{x}} - \mu)\\
&= \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \cdot f_{\text{Sum}}(\bar{\mathbf{e}}),\\
&⑪ : -\frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \cdot f_{\text{Sum}}(\bar{\mathbf{e}}),\\
&⑫ : -\frac{\gamma}{M\sqrt{\sigma^2 + \varepsilon}} \cdot f_{\text{Sum}}(\bar{\mathbf{e}}) \cdot 1_M,\\
&⑬ : \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}}\bar{\mathbf{e}}\\
&-\frac{\gamma}{M}\left(\sigma^2 + \varepsilon\right)^{-3/2} \cdot f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}}) \cdot (\check{\mathbf{x}} - \mu),\\
&⑭ : f_{\text{Sum}}(\bar{\mathbf{e}}),\\
&⑮ : f_{\text{Sum}}(\mathbf{xe}\bar{\mathbf{e}}) = f_{\text{Sum}}\left(\frac{\check{\mathbf{x}} - \mu}{\sqrt{\sigma^2 + \varepsilon}} \odot \bar{\mathbf{e}}\right).
\end{aligned}
\tag{A.1}
$$

The fact that $f_{\text{Sum}}(\check{\mathbf{x}} - \mu) = 0$ is used when deriving the result of ⑩. Finally, the backward propagation error arriving at the input of the batch-normalization function is given as the resulting sum of ⑫ and ⑬, which can be written as

$$
\check{\mathbf{e}} = \frac{\gamma}{M\sqrt{\sigma^2 + \varepsilon}}\left(M \cdot \bar{\mathbf{e}} - f_{\text{Sum}}(\bar{\mathbf{e}}) \cdot 1_M - \frac{f_{\text{Sum}}((\check{\mathbf{x}} - \mu) \odot \bar{\mathbf{e}})}{\sigma^2 + \varepsilon} \cdot (\check{\mathbf{x}} - \mu)\right). \tag{A.2}
$$

Data Availability

The data is available at http://home.konkuk.ac.kr/~ecyoon/DATA_PAPER/Data_for_Figures.pdf.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2018R1D1A1B07051392, No. NRF-2018R1D1A1B07050232, No. NRF-2021R1F1A1047578).

References

[1] A. Abdi, H. Zhang, and C. Tepedelenlioglu, "A unified approach to the performance analysis of speed estimation techniques in mobile communication," *IEEE Transactions on Communications*, vol. 56, no. 1, pp. 126–135, 2008.

[2] K. E. Baddour and N. C. Beaulieu, "Robust doppler spread estimation in nonisotropic fading channels," *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 2677–2682, 2005.

[3] A. Dogandzic and B. Zhang, "Estimating Jakes' Doppler power spectrum parameters using the whittle approximation," *IEEE Transactions on Signal Processing*, vol. 53, no. 3, pp. 987–1005, 2005.

[4] K. E. Baddour and N. C. Beaulieu, "Nonparametric Doppler spread estimation for narrowband wireless channels," *IEEE Wireless Communications and Networking (WCNC) Conference*, vol. 2, pp. 953–958, 2003.

[5] H. Zhang and A. Abdi, "Nonparametric mobile speed estimation in fading channels: performance analysis and experimental results," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1683–1692, 2009.

[6] E. Yoon, J. Kim, and U. Yun, "Doppler spread estimation for an OFDM system with a Rayleigh fading channel," *IEICE Transactions on Communications*, vol. E101.B, no. 5, pp. 1328–1335, 2018.

[7] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pp. 511–516, Geneva, Switzerland, 2013.

[8] R. Safdari and M.-S. Moin, "A hierarchical feature learning for isolated farsi handwritten digit recognition using sparse autoencoder," in *2016 Artificial Intelligence and Robotics (IRANOPEN)*, pp. 67–71, Qazvin, Iran, 2016.

[9] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 341–346, Monticello, IL, USA, September 2016.

[10] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[11] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[12] S. Dorner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE Journal of*

Selected Topics in Signal Processing, vol. 12, no. 1, pp. 132–143, 2018.

[13] M. Kim, W. Lee, and D.-H. Cho, "A novel PAPR reduction scheme for OFDM system based on deep learning," *IEEE Communications Letters*, vol. 22, no. 3, pp. 510–513, 2018.

[14] U. Challita, L. Dong, and W. Saad, "Proactive resource management for LTE in unlicensed spectrum: a deep learning perspective," *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4674–4689, 2018.

[15] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852–855, 2018.

[16] X. Gao, S. Jin, C.-K. Wen, and G. Y. Li, "ComNet: combination of deep learning and expert knowledge in OFDM receivers," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2627–2630, 2018.

[17] M. Zamanipour, "A survey on deep-learning based techniques for modeling and estimation of massive MIMO channels," 2019, https://arxiv.org/abs/1910.03390.

[18] Z. Qin, H. Ye, G. Y. Li, and B.-H.-F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, 2019.

[19] Z.-M. Liu, C. Zhang, and S. Y. Philip, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–7327, 2018.

[20] D. Hu, Y. Zhang, L. He, and J. Wu, "Low-complexity deep-learning-based DOA estimation for hybrid massive mimo systems with uniform circular arrays," *IEEE Wireless Communications Letters*, vol. 9, no. 1, pp. 83–86, 2020.

[21] L. Wu, Z.-M. Liu, and Z.-T. Huang, "Deep convolution network for direction of arrival estimation with sparse prior," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1688–1692, 2019.

[22] D. Burghal, N. A. Abbasi, and A. F. Molisch, "A machine learning solution for beam tracking in mmWave systems," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 173–177, Pacific Grove, CA, USA, 2019.

[23] K. N. R. S. V. Prasad, E. Hossain, V. K. Bhargava, and S. Mallick, "Analytical approximation-based machine learning methods for user positioning in distributed massive MIMO," *IEEE Access*, vol. 6, pp. 18431–18452, 2018.

[24] S. Tsuchida, T. Takahashi, S. Ibi, and S. Sampei, "Machine learning-aided indoor positioning based on unified fingerprints of Wi-Fi and BLE," in *Proceedings of APSIPA Annual Summit and Conf.*, pp. 1468–1472, Lanzhou, China, Nov. 2019.

[25] Y. Kim and T. Moon, "Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, 2016.

[26] B. Kihei, J. A. Copeland, and Y. Chang, "Automotive Doppler sensing: the Doppler profile with machine learning in vehicle-to-vehicle networks for road safety," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, Japan, 2017.

[27] T. Kim, K. Ko, I. Hwang, D. Hong, S. Choi, and H. Wang, "RSRP-based Doppler shift estimator using machine learning in high-speed train systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 371–380, 2021.

[28] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pp. 26–33, Cambridge, July 2001, https://goo.gl/KNZMEA.

[29] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspective," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1798–1828, 2013.

[30] A. Geron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly, 2017, Chapter 1.

[31] S. Medawar, P. Handel, and P. Zetterberg, "Approximate maximum likelihood estimation of Rician K-factor and investigation of urban wireless measurements," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2545–2555, 2013.

[32] K. E. Baddour and N. C. Beaulieu, "Nonparametric Doppler spread estimation for flat fading channel," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*, New Orleans, LA, USA, 2003.

[33] Y. Choi, P. J. Voltz, and F. A. Cassara, "On channel estimation and detection for multicarrier signals in fast and selective Rayleigh fading channels," *IEEE Transactions on Communications*, vol. 49, no. 8, pp. 1375–1387, 2001.

[34] S. Song and A. C. Singer, "Pilot-aided OFDM channel estimation in the presence of the guard band," *IEEE Transactions on Communications*, vol. 55, no. 8, pp. 1459–1465, 2007.

[35] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems*, Kluwer Academic/Plenum, New York, 2nd edition, 2000.

[36] C. Tepedelenlio and G. B. Giannakis, "On velocity estimation and correlation properties of narrow-band mobile communication channels," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 4, pp. 1039–1052, 2001.

[37] A. Abdi and M. Kaveh, "Parametric modeling and estimation of the spatial characteristics of a source with local scattering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 2821–2824, Orlando, FL, 2002.

[38] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis, Ppper Saddle River*, Prentice Hall, NJ, 1997.

[39] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[41] P. Ramachandran, Z. Barret, and Q. V. Le, "Searching for activation functions," 2017, https://arxiv.org/abs/1710.05941v2.

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imageNet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, Santiago, Chile, 2015.

[44] A. Karpathy, *Hacker's Guide to Neural Networks*, Andrej Karpathy's Blog, 2014, http://karpathy.github.io/neuralnets/.

[45] CS231n, "Convolutional neural networks for visual recognition," https://cs231n.github.io/.

[46] O. H. Rodriguez, L. Fernandez, and M. Jorge, "A semiotic reflection on the didactics of the chain rule," *The Mathematics Enthusiast*, vol. 7, no. 2, pp. 321–332, 2010.

[47] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for Machine Learning*, MIT Press, Cambridge, 2012.

[48] F. Kratzert, *Understanding the Backward Pass through Batch Normalization Layer*, Frederik Kratzert's blog, 2016, https://kratzert.github.io/2016/02/12/understanding-the-gradient-flow-through-the-batch-normalization-layer.html.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9627776, 11 pages
https://doi.org/10.1155/2021/9627776

Research Article

Air Quality Prediction Model Based on Spatiotemporal Data Analysis and Metalearning

Kejia Zhang,[1] Xu Zhang,[1] Hongtao Song,[1] Haiwei Pan,[1] and Bangju Wang[2]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin /150001, China*
[2]*College of Science, Huazhong Agricultural University, Wuhan /430070, China*

Correspondence should be addressed to Hongtao Song; songhongtao@hrbeu.edu.cn

Received 26 June 2021; Accepted 16 August 2021; Published 28 August 2021

Academic Editor: Xiao Zhang

With the continuous improvement of people's quality of life, air quality issues have become one of the topics of daily concern. How to achieve accurate predictions of air quality in a variety of complex situations is the key to the rapid response of local governments. This paper studies two problems: (1) how to predict the air quality of any monitoring station based on the existing weather and environmental data while considering the spatiotemporal correlation among monitoring stations and (2) how to maintain the accuracy and stability of the forecast even when the available data is severely insufficient. A prediction model combining Long Short-Term Memory networks (LSTM) and Graph Attention (GAT) mechanism is proposed to solve the first problems. A metalearning algorithm for the prediction model is proposed to solve the second problem. LSTM is used to characterize the temporal correlation of historical data and GAT is used to characterize the spatial correlation among all the monitoring stations in the target city. In the case of insufficient training data, the proposed metalearning algorithm can be used to transfer knowledge from other cities with abundant training data. Through testing on public data sets, the proposed model has obvious advantages in accuracy compared with baseline models. Combining with the metalearning algorithm, it gives a much better performance in the case of insufficient training data.

1. Introduction

Because of the increasingly serious air pollution all over the world, air quality has become one of the most socially concerned issues. In many countries, air quality has become a key indicator to measure the happiness index of residents. In order to achieve real-time monitoring of air quality, almost all countries have arranged a large number of air quality monitoring stations in major cities. Besides, more and more mobile portable monitoring devices are participating in air quality monitoring [1, 2]. Although many monitoring methods have been applied, it is still extremely challenging to make accurate predictions of air quality. Especially in the case of insufficient monitoring data or poor data quality, it is more difficult to maintain the accuracy and stability of the prediction model.

Air quality is affected by a variety of complex factors [3–6], including meteorological factors, industrial factors, fuel factors, traffic factors, and other human activity factors. The development of related monitoring equipment leads the collection of air quality data more and more comprehensive [7]. With the application of a series of spatiotemporal prediction models [8], air quality prediction has made considerable progress. Time correlation refers to the impact of historical monitoring data on future data, and spatial correlation refers to the mutual influence among adjacent monitoring stations. Most of the existing research works [3, 4] focus on establishing prediction models based on time correlation, while there are obvious shortcomings in the study of spatial correlation. The reason for this phenomenon is because the diffusion of air pollutants is affected by various factors such as geographical location, wind direction, wind speed, air pressure, and air humidity. The impact of each factor on the relevance of different regions is difficult to accurately model. In this paper, we propose a spatiotemporal model for air quality prediction. The proposed model combines Long and Short-Term Memory networks (LSTM) and Graph ATtention (GAT) mechanism [9], where LSTM is used to

capture the correlation in the time domain and GAT is used to model the spatial correlation among different regions.

In recent years, many deep learning models [3, 4, 10–13] have achieved good results in air quality prediction. However, the accuracy of these predicting models highly depends on the sufficiency of training data. In reality, a lack of sufficient training data is the most common situation. As we all know, air quality monitoring in developing countries mainly depends on monitoring stations arranged by the government. There are few or no such stations in small cities and towns. Insufficient training data makes it difficult for the existing prediction models to achieve accurate results in these small cities and towns. Even in large cities, government-monitoring stations are very sparse. Although there are some unofficial monitoring devices that can provide data as a supplement, the data collected by these simple monitoring devices are often of poor quality with large amounts of various dirty data and missing values. Therefore, making accurate predictions based on insufficient training data is a realistic and challenging problem. Transfer learning (metalearning) [14] is currently the most effective method to solve this problem. Some transfer learning models [15–17] have been proposed to predict air quality with insufficient data. However, these methods require a strong similarity between the source domain and the target domain. Different cities and towns (especially large cities and small towns) have huge differences in pollution levels, climate, pollutant diffusion conditions, and density of monitoring sites. This makes it difficult for the existing transfer learning technology to successfully transfer the knowledge acquired in large cities to the air quality prediction of small and medium cities. To meet these challenges, based on the proposed prediction model, we give a metalearning algorithm for knowledge transfer among cities with huge differences.

The main contributions of this paper are as follows:

(i) Proposing a spatiotemporal model by combining LSTM and GAT for accurate air quality prediction

(ii) Designing a metalearning algorithm for the proposed model, which can transfer knowledge among different cities and make an accurate prediction in case of insufficient training data

(iii) Verifying the advantages of the proposed model and meta-learning algorithm in the aspect of prediction accuracy through a large number of experiments

The rest of this paper is organized as Section 2 introduces some related research works in the area of air quality prediction, transfer learning, and metalearning; Section 3 gives the definition of the problems; the proposed prediction model and metalearning algorithm are introduced in Section 4; after showing the experimental results to prove the effectiveness of the proposed model and metalearning algorithm in Section 5, Section 6 summarizes the whole paper.

2. Related Works

This section will briefly present the related research works in the area of air quality prediction, transfer learning, and metalearning.

2.1. Air Quality Prediction. The machine learning models for air quality prediction can be divided into two categories: basic learning models and deep learning models. Basic learning models include linear regression, supporting vector regression, random forest, and LightGBM. Land Use Regression (LUR) [5, 6] makes air quality predictions through a linear regression model that takes into account multiple factors like regional population level, traffic condition, and land use condition. LUR does not consider the complicated spatiotemporal correlation of air pollution data, so the accuracy of prediction is poor. Later, the basic time series model autoregressive integrated moving average model (ARIMA) [18] appeared, which was used for time series forecasting with strong periodicity. However, it does not perform well for complex weather conditions. Random forest [19], LightGBM [20], deep learning methods have become widely used methods in air pollution prediction. Later, in order to further improve the accuracy of prediction, Zheng et al. [21] proposed U-Air, which uses a spatial classifier based on an artificial neural network (ANN) and a temporal classifier based on the linear-chain conditional random field (CRF) to capture temporal and spatial characteristics. Convolutional neural networks (CNN) are used to process data from Euclidean structures. For example, they are very effective in the field of image recognition, and it is impractical to use CNN directly to capture the spatial relationships between monitoring stations for sparse graph structures consisting of monitoring stations. The ConvLSTM model proposed in [22] combines CNN and LSTM to characterize the spatiotemporal relationship between monitoring stations, and it is still applicable to the spatial relationship in Euclidean space. The emergence of Graph Convolutional Networks (GCN) [23, 24] has made up for the deficiencies of CNN and is widely used in traffic data. GCN has realized the full use of the traffic network. GAT [9] are proposed on the basis of GCN, using an attention mechanism, and are good at capturing dynamic relationships between nodes. The ST-GAT model proposed by Zhang et al. [25] can dynamically capture the dynamic dependencies in the traffic network, making the traffic speed prediction results more advanced than existing models.

2.2. Transfer Learning and Meta-Learning. To improve the practicality of air quality prediction models, the obstacles caused by insufficient data must be resolved. Transfer learning can be divided into three categories according to the difference in source domains and target domains and tasks, namely, inductive transfer learning, transitive transfer learning, and unsupervised transfer learning [14]. In recent years, transfer learning combined with deep neural networks (DNN) has been widely used. The VGG model proposed in the image field [26], with the help of this model, can achieve fast and accurate model training under a small number of sets. Unlike image data, air quality data are more complex in spatial and temporal distribution. Hu et al. [27] proposed a DNN-based sharing model that fused multi-source wind speed data together to solve the problem of insufficient wind farm data. However, this model does not

provide a solution to the knowledge transfer of spatially related data.

Metalearning [28–30] can quickly initialize the model by learning knowledge in multiple different learning tasks in order to widely adapt to a variety of situations. Literature [29] firstly proposes the concept of metalearning, also known as learning. The goal is to train a metalearning model on multiple learning tasks, so as to use a small number of training samples to solve new learning tasks. A model-independent metalearning algorithm MAML is proposed in [28]. MAML deals with the situation of insufficient training data by transferring data and models among multiple learning tasks. Each update step consists of multitask pretraining, model migration, target task training, and model parameter synchronization. Unlike previous metalearning methods, MAML uses gradients to update model and does not introduce additional parameters. Literature [27] proposes a MAML-based spatiotemporal prediction model, which is used for urban traffic prediction and water quality prediction by transferring knowledge among multiple cities.

2.3. Summary of Related Works. Through the introduction of the above related works, it can be seen that the existing air quality prediction methods rarely consider the spatial relationship between multiple monitoring stations. A few spatiotemporal prediction models lack the ability to dynamically model spatial correlation based on weather and other related factors. The only methods that can dynamically model spatial correlation do not consider how to deal with insufficient training data. Some existing methods in the area of transfer learning and metalearning can solve the insufficient-training-data situation to a certain extent by transferring the knowledge from other source domains, but these methods lack the ability to adapt to the air quality spatiotemporal prediction models and cannot be directly applied to the scenarios targeted in this article. For this reason, this paper proposes a spatiotemporal model for air quality prediction and a metalearning algorithm for this model. The prediction model can dynamically and accurately model the temporal and spatial correlation in air quality prediction. The metalearning algorithm is used to establish a more accurate prediction model in the case of insufficient training data. As far as we know, it is the first time that metalearning has been used for air quality prediction.

3. Problem Formulation

This paper will solve two problems: prediction problem and transfer learning problem. The prediction problem is how to build a prediction model for the target pollutant in the city with sufficient training data. The transfer learning problem is how to build a prediction model in the target city with insufficient training data, given the source cities with sufficient data. The symbols used in this paper are given in Table 1.

3.1. Prediction Problem. Suppose that there is a set of urban monitoring stations $\boldsymbol{S} = \{s_1, s_2, \cdots, s_n\}$ in the target city. We use a fixed time interval while counting historical data and making predictions. The prediction problem is building a model to predict the concentration of a certain pollutant sampled by a specified monitoring station in the future. The target air pollutant can be one of PM2.5, PM10, SO_2, NO_2, O_3, CO, AQI (can be regarded as a comprehensive pollutant). Suppose that the current time is t. The input of the prediction model contains (1) a specified monitoring station, (2) the historical monitoring data of the target pollutant sampled from time $t-k$ to time t, (3) the historical weather information from time $t-k$ to time t, and (4) the weather forecast information from time $t+1$ to time $t+l$. The output of the prediction model is the predicted value of the target pollutant sampled by the specified station from time $t+1$ to time $t+l$. In practical applications, we usually set $k+1=2l$.

For a monitoring station $s \in \boldsymbol{S}$, define the historical data vector of s as $\boldsymbol{x}_s = (x_s^{t-k}; x_s^{t-k+1}; \cdots; x_s^t)$, where x_s^i $(t-k \le i \le t)$ is the concentration of the target pollutant sampled by station s at time i. The historical data vectors of all monitoring stations can be represented by a matrix $\boldsymbol{X} = (\boldsymbol{x}_{s_1}, \boldsymbol{x}_{s_2}, \cdots, \boldsymbol{x}_{s_n})$.

The weather information used by the prediction model includes temperature, humidity, pressure, wind direction, and wind speed. The historical weather dataset of the target city is expressed as $\boldsymbol{W}_h = (\boldsymbol{w}^{t-k}, \boldsymbol{w}^{t-k+1}, \cdots, \boldsymbol{w}^t)$, in which $\boldsymbol{w}^i$ $(t-k \le i \le t)$ represents the vector of the above weather indexes sampled at time i. The weather forecast dataset of the target city is expressed as $\boldsymbol{W}_f = (\boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+2}, \cdots, \boldsymbol{w}^{t+l})$, in which $\boldsymbol{w}^i (t+1 \le i \le t+l)$ represents the forecast value of the above weather indexes at time i. Usually, $\boldsymbol{W}_h$ and $\boldsymbol{W}_f$ are given by the meteorological department.

Given the target monitoring station $s \in \boldsymbol{S}$, our goal is to predict the concentration of the target pollutant sampled by s from time $t+1$ to time $t+l$, which can be expressed as a vector $\boldsymbol{y}_s = (y_s^{t+1}; y_s^{t+2}; \cdots, y_s^{t+l})$. Suppose that f_θ with parameter θ is the model we build, so we have

$$\hat{\boldsymbol{y}}_s = f_\theta(\boldsymbol{X}, \boldsymbol{W}_h, \boldsymbol{W}_f, s), \tag{1}$$

where $\hat{\boldsymbol{y}}_s$ is the predicion of $\boldsymbol{y}_s$.

Let $\boldsymbol{D}_{tc}$ be the training dataset of the target city. $\boldsymbol{D}_{tc}$ contains the historical monitoring data of all monitoring stations, the historical weather data, and the historical weather forecast data collected from the target city over a period of time. The prediction problem can be formally defined as how to build an accurate prediction model f_θ based on $\boldsymbol{D}_{tc}$.

3.2. Transfer Learning Problem. In addition to constructing the prediction model, another important issue to be solved in this paper is how to make accurate predictions when there is little training data. In this case, we will transfer knowledge from the source cities with sufficient training data to the target city with insufficient data. Suppose that we have m source cities with sufficient training data. Let $\boldsymbol{D}_{sc}^1$, $\boldsymbol{D}_{sc}^2$, ..., $\boldsymbol{D}_{sc}^m$ be the training datasets collected from the source cities, respectively. Let $\boldsymbol{D}_{tc}$ be the insufficient training dataset collected from the target city. The transfer learning problem

Table 1: Symbols and explanations.

Symbol	Meaning
$\boldsymbol{S} = \{s_1, s_2, \cdots, s_n\}$	Set of monitoring stations in the target city
k	Length of the historical data
t	Current time
l	Length of the forecast window
x_s^i	Concentration of target pollutant monitored by station s at time i (past)
$\boldsymbol{x}_s = \left(x_s^{t-k}; x_s^{t-k+1}; \cdots, x_s^t\right)$	Historcical data vector of station s
$\boldsymbol{X} = \left(\boldsymbol{x}_{s_1}, \boldsymbol{x}_{s_2}, \cdots, \boldsymbol{x}_{s_n}\right)$	Historcical data vectors of all stations
$\boldsymbol{w}^i (t-k \le i \le t)$	Historical weather of the target city at time i
$\boldsymbol{W}_h = \left(\boldsymbol{w}^{t-k}, \boldsymbol{w}^{t-k+1}, \cdots, \boldsymbol{w}^t\right)$	Historical weather dataset
$\boldsymbol{w}^i (t+1 \le i \le t+l)$	Weather forecast of the target city at time i
$\boldsymbol{W}_f = \left(\boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+2}, \cdots, \boldsymbol{w}^{t+l}\right)$	Weather forecast dataset
$\boldsymbol{T} = \left\{\boldsymbol{D}_{\boldsymbol{sc}}^{\boldsymbol{1}}, \boldsymbol{D}_{\boldsymbol{sc}}^{\boldsymbol{2}}, \cdots, \boldsymbol{D}_{\boldsymbol{sc}}^{\boldsymbol{m}}\right\}$	Set of training dataset from source cities
$\boldsymbol{D}_{tc}$	Training dataset from the target city
$y_s^i (t+1 \le i \le t+l)$	Concentration of target pollutant monitored by station s at time i (future)
$\boldsymbol{y}_s = \left(y_s^{t+1}; y_s^{t+2}; \cdots, y_s^{t+l}\right)$	Concentration of target pollutant at station s in the future
f_θ	Prediction model with parameter θ
$\mathscr{L}_{\boldsymbol{B}}(f_\theta)$	Loss of model f_θ on training batch $\boldsymbol{B}$
$\widehat{\boldsymbol{y}}_s = \left(\widehat{y}_s^{t+1}; \widehat{y}_s^{t+2}; \cdots, \widehat{y}_s^{t+l}\right)$	Predicted value of $\boldsymbol{y}_s$.
r	Influence radius of monitoring stations
G_r	Directed graph built on all the monitoring stations
$N_r(s)$	$N_r(s) = \{s\} \cup \left\{s' \mid \left\langle s', s\right\rangle \in G_r\right\}$
$\boldsymbol{h}_s^i$	Output vector of the LSTM unit on station s at time i
$\boldsymbol{c}_s^i$	Cell state vector of the LSTM unit on station s at time i
$\boldsymbol{z}_s^i$	Output vector of GAT on station s at time i
$\alpha_{s',s}$	GAT's similarity score between node s' and s
$\widehat{\alpha}_{s',s}$	Weight of the edge $\left\langle s', s\right\rangle$ in G_r calculated by GAT

is defined as how to build an accurate prediction model f_θ for the target city based on $\boldsymbol{D}_{\boldsymbol{tc}} \cup \boldsymbol{D}_{\boldsymbol{sc}}^{\boldsymbol{1}} \cup \boldsymbol{D}_{\boldsymbol{sc}}^{\boldsymbol{2}} \cup \cdots \cup \boldsymbol{D}_{\boldsymbol{sc}}^{\boldsymbol{m}}$.

4. Methodology

4.1. Monitoring Station Graph. In order to measure the mutual influence among different monitoring stations, we initially model all monitoring stations as a directed graph G_r. Monitoring stations are represented by the nodes (vertices) in G_r. Given two nodes s_i and s_j, there are directed edges $\langle s_i, s_j\rangle$ and $\langle s_j, s_i\rangle$ if the Euclidean distance between s_i and s_j is less than or equal to r. r is the influence radius of monitoring stations, i.e., the maximum range affected by the pollutant in the diffusion process. As shown in Figure 1, by setting $r = 20$km, we get the graph among 34 monitoring stations located in Beijing, China. The weights of the edges in G_r will be calculated by GAT mechanism and change over time. Hereinafter, we use the term "node" to refer to monitoring station and define set $N_r(s)$ as

$$N_r(s) = \{s\} \cup \left\{s' \mid \left\langle s', s\right\rangle \in G_r\right\}. \tag{2}$$

4.2. Air Quality Prediction Model. To solve the prediction problem, we propose a spatiotemporal prediction model (referred as GAT-LSTM) as shown in Figure 2. The model is built by a recurrent neural network incorporating graph attention mechanism, which means that it has encoder-decoder structure. The encoder is used to embed historical data, and the decoder is used to generate the predicted value in the future. It uses LSTM to model time correlation of a

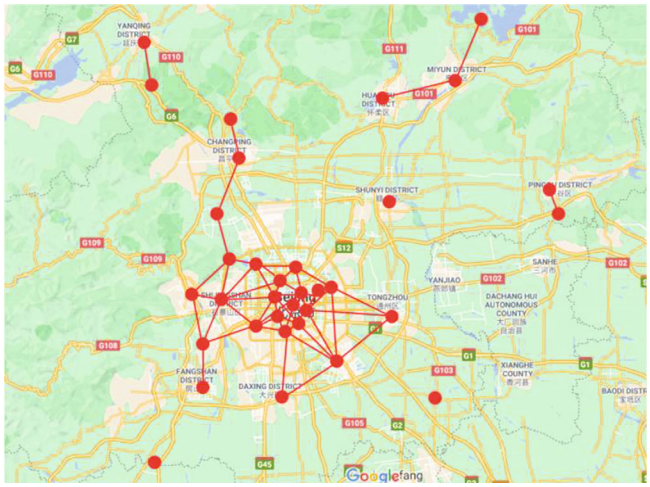


FIGURE 1: The Graph of 34 monitoring stations in Beijing ($r = 20$km).

node's own data and uses GAT to capture spatial correlation among nodes.

Suppose that the current time is t. Take node s as an example. In the encoding phase, we use $k+1$ LSTM units to receive s's historical data from time $t-k$ to time t, and form them into a recurrent neural network. For time $i\,(t-k \le i \le t)$, the input of the corresponding LSTM unit is $(x_s^i; \boldsymbol{w}^i)$, where x_s^i is the concentration of target pollutant monitored by s at time i, and $\boldsymbol{w}^i$ is the historical weather data at time i. Let $\boldsymbol{h}_s^i$ and $\boldsymbol{c}_s^i$ be the output vector (blue lines in Figure 2) and the cell state vector (gray lines in Figure 2) of the LSTM unit at time i, respectively. Unlike the traditional approach passing $\boldsymbol{h}_s^i$ and $\boldsymbol{c}_s^i$ directly to the next LSTM unit, we pass $\boldsymbol{h}_s^i$ to GAT to find spatial correlation among different nodes. Let $\boldsymbol{z}_s^i$ be the output vector of GAT for node s at time i. In the end, $\boldsymbol{z}_s^i$ and $\boldsymbol{c}_s^i$ are passed to the next LSTM unit. The structure of LSTM unit is as shown in Figure 3.

In the decoding phase, l LSTM units are used to generate $(\hat{y}_s^{t+1}; \hat{y}_s^{t+2}; \cdots, \hat{y}_s^{t+l})$, i.e., the predicted concentration of the target pollutant sampled by node s. For time $j\,(t+1 \le j \le t+l)$, the input of the corresponding LSTM unit is $(y_s^{j-1}; \boldsymbol{w}^j)$, where y_s^{j-1} is the prediction of the previous moment and $\boldsymbol{w}^j$ is the weather forecast data at time j. As with the coding phase, we pass the LSTM's output vector $\boldsymbol{h}_s^j$ to GAT to generate vector $\boldsymbol{z}_s^j$. In addition to being passed to the next LSTM unit, $\boldsymbol{z}_s^j$ is also passed to a Feedforward Neural Network (FNN) to generate the output y_s^j.

Base on the monitoring station graph G_r, we use a GAT to model the spatial relationship among different nodes. In GAT, each node uses the attention mechanism [31] to collect information from neighbor nodes (weighting and summing the feature vectors of neighbor nodes) and uses the collected information to update its own feature vector. Unlike GCN, the weight of an edge in GAT is calculated based on the similarity of the feature vectors of the two corresponding nodes and changes dynamically with the change of the node's data. GAT is very sensitive to the changes of the spatial correlation among nodes caused by weather factors such as wind speed and wind direction.

The GAT mechanism can be demonstrated by Figure 4. At any time, the input of GAT is the output vectors of all nodes' LSTM units, i.e., $\{\boldsymbol{h}_{s_1}, \boldsymbol{h}_{s_2}, \cdots, \boldsymbol{h}_{s_n}\}$. The output of GAT is $\{\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \cdots, \boldsymbol{z}_{s_n}\}$. Each $\boldsymbol{z}_s$ is passed to the next LSTM unit on the corresponding node as a hidden state vector. To get $\boldsymbol{z}_s$, for each $s' \in N_r(s)$, GAT firstly calculates the similarity score between node s' and s by

$$\alpha_{s',s} = \boldsymbol{v}^{\mathrm{T}} \tanh\left(U_1 \boldsymbol{h}_{s'} + U_2 \boldsymbol{h}_s\right), \tag{3}$$

where vector $\boldsymbol{v}$ and matrix U_1 and U_2 are the parameters that need to be learned. Then, $\widehat{\alpha}_{s',s}$ is calculated by normalizing all the $\alpha_{s',s}$ through the softmax layer:

$$\widehat{\alpha}_{s',s} = \mathrm{softmax}\left(\alpha_{s',s}\right) = \frac{\exp\left(\alpha_{s',s}\right)}{\sum_{u \in N_r(s)} \exp\left(\alpha_{u,s}\right)}. \tag{4}$$

$\widehat{\alpha}_{s',s}$ can be seen as the weight of the edge $\langle s', s\rangle$ in G_r. Finally, $\boldsymbol{z}_s$ is calculated by a weighted summation of all its neighbors' $\boldsymbol{h}$, i.e.,

$$\boldsymbol{z}_s = \sum_{s' \in N_r(s)} \widehat{\alpha}_{s',s} \boldsymbol{h}_{s'}. \tag{5}$$

4.3. Metalearning Algorithm. To solve the transfer learning problem, we propose a metalearning algorithm named MetaGAT-LSTM (given by Algorithm 1) for training the GAT-LSTM model in the target city with insufficient training data. The algorithm will build an accurate prediction model by transferring knowledge from source cities with sufficient data. It uses a modified version of Model-Agnostic Meta-Learning (MAML) [28] as the parameter learning method.

Let $\boldsymbol{T} = \{\boldsymbol{D}_{sc}^1, \boldsymbol{D}_{sc}^2, \cdots, \boldsymbol{D}_{sc}^m\}$ be the set of datasets from source cities. Define the distribution over $\boldsymbol{T}$ as $\mathscr{P}(\boldsymbol{T})$, in which the probability of choosing dataset $\boldsymbol{D} \in \boldsymbol{T}$ is

$$\Pr\left(\boldsymbol{D}\right) = \frac{|\boldsymbol{D}|}{\Sigma_{\boldsymbol{D}' \in \boldsymbol{T}} |\boldsymbol{D}'|}. \tag{6}$$

Let f_θ with parameter θ be the prediction model at the beginning of each training iteration. At first, with respect to $\mathscr{P}(\boldsymbol{T})$, we sample k datasets $\boldsymbol{D}_1, \boldsymbol{D}_2, \cdots, \boldsymbol{D}_k$ from $\boldsymbol{T}$ with replacement (Line 3 in Algorithm 1). Then, get the next training batches $\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_k$ from $\boldsymbol{D}_1, \boldsymbol{D}_2, \cdots, \boldsymbol{D}_k$, respectively (Line 4 in Algorithm 1) and get the next training batch $\boldsymbol{B}_0$ from $\boldsymbol{D}_{tc}$ (Line 5 in Algorithm 1). The model's parameter is updated in two steps. In the first step (Line 6 ~ 10 in Algorithm 1), for each $\boldsymbol{B}_i\,(1 \le i \le k)$, we get the first-step adapted parameter θ_i by

$$\theta_i = \theta - \beta \nabla \mathscr{L}_{\boldsymbol{B}_i}(f_\theta). \tag{7}$$

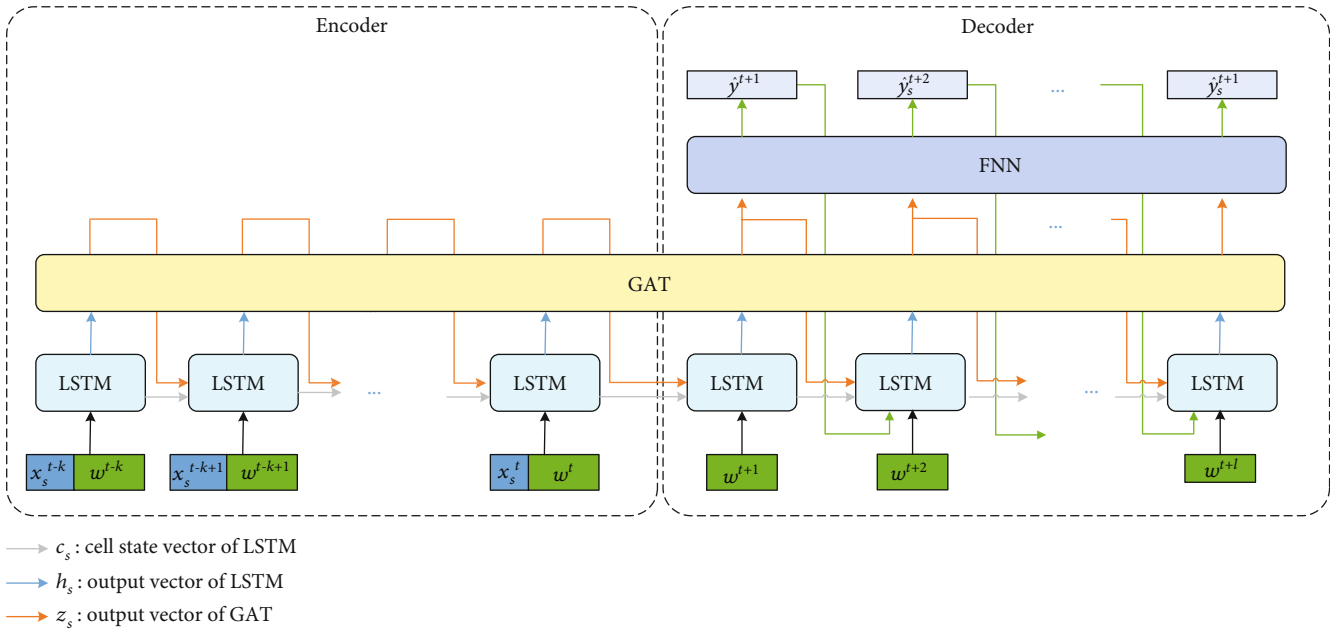


Figure 2: The air quality prediction model GAT-LSTM.

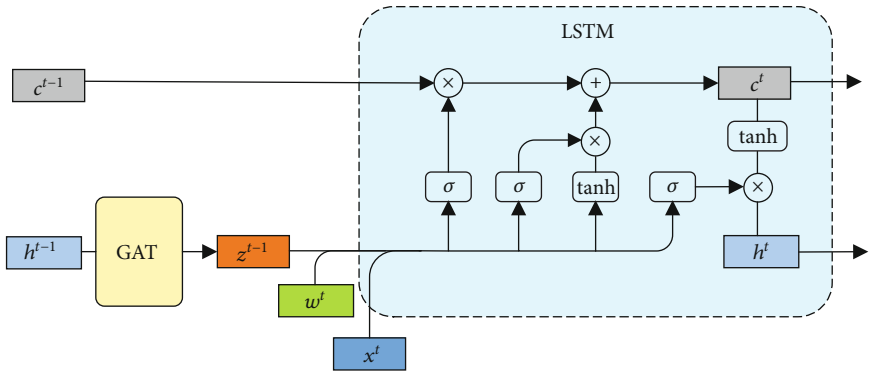


Figure 3: The LSTM unit in GAT-LSTM.

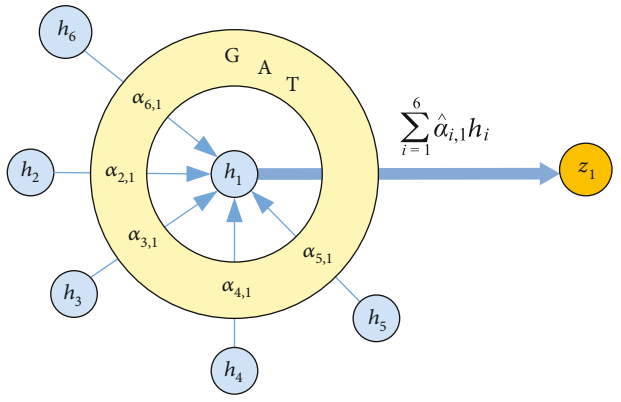


Figure 4: The GAT mechanism for node 1 (suppose that node 2 ~ 6 are 1's neighbors in G_r).

where $\mathscr{L}_{\boldsymbol{B}_i}(f_\theta)$ is the loss of the original model f_θ on training batch $\boldsymbol{B}_i$, $\nabla\mathscr{L}_{\boldsymbol{B}_i}(f_\theta)$ is gradient of $\mathscr{L}_{\boldsymbol{B}_i}(f_\theta)$ and β is the learning rate in the first step. In the second step (Line 11 in Algorithm 1), we get the second-step adapted parameter by

$$\theta = \theta - \gamma \sum_{i=1}^{k} \nabla\mathscr{L}_{\boldsymbol{B}_0}\left(f_{\theta_i}\right), \tag{8}$$

where $\mathscr{L}_{\boldsymbol{B}_0}(f_{\theta_i})$ is the loss of first-step adapted model f_{θ_i} on training batch $\boldsymbol{B}_0$ and γ is the learning rate in the second step.

5. Experimental Results and Analysis

5.1. Dataset. We use real datasets collected from four cities in China (Beijing, Tianjin, Shenzhen, Guangzhou) to verify the effectiveness and efficiency of the proposed model and meta-learning algorithm. These cities are very different in

Input: $\boldsymbol{T} = \{\boldsymbol{D}_{sc}^1, \boldsymbol{D}_{sc}^2, \cdots, \boldsymbol{D}_{sc}^m\}$: The set of the training datasets from source cities;
$\boldsymbol{D}_{tc}$: Datasets from target city;
$\mathscr{P}(\boldsymbol{T})$: Distribution over $\boldsymbol{T}$;
β, γ: Learning rates
Output: f_θ: The GAT-LSTM model for the target city
1. Randomly initialize θ
2. **While** not done **do**:
3. Sample k datasets $\boldsymbol{D}_1, \boldsymbol{D}_2, \cdots, \boldsymbol{D}_k$ from $\boldsymbol{T}$ with replacement w.r.t. $\mathscr{P}(\boldsymbol{T})$
4. Get next training batches $\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_k$ from $\boldsymbol{D}_1, \boldsymbol{D}_2, \cdots, \boldsymbol{D}_k$, respectively
5. Get next training batch $\boldsymbol{B}_0$ from $\boldsymbol{D}_{tc}$
6. **For** $\boldsymbol{B}_i$ in $\{\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_k\}$ **do**:
7. Calculate $\nabla \mathscr{L}_{\boldsymbol{B}_i}(f_\theta)$ with respect to $\boldsymbol{B}_i$
8. Calculate first-step adapted parameter $\theta_i \longleftarrow \theta - \beta \nabla \mathscr{L}_{\boldsymbol{B}_i}(f_\theta)$
9. Calculate $\nabla \mathscr{L}_{\boldsymbol{B}_0}(f_{\theta_i})$ with respect to $\boldsymbol{B}_0$
10. **End for**
11. Calculate second-step adapted parameter $\theta \longleftarrow \theta - \gamma \sum_{i=1}^{k} \nabla \mathscr{L}_{\boldsymbol{B}_0}(f_{\theta_i})$
12. **End while**

Algorithm 1: MetaGAT-LSTM.

geographic coordinates, city size, population density, etc., resulting in very different air quality distributions. For example, the air pollution situation in Beijing and Tianjin is much more serious than that in Shenzhen and Guangzhou. Each dataset contains the air quality data (from all monitoring stations), weather data, and weather forecast data collected from a city within one year. The period of data sampling is one hour. Taking the dataset of Beijing as an example, the air quality data contains the concentration of six major pollutants (PM2.5, PM10, SO2, NO2, O3, CO) and AQI sampled by 36 monitoring stations within one year. The weather data contains basic weather, temperature, humidity, air pressure, wind speed, and wind direction collected within one year. Weather forecast data contains the forecast value of the above weather indexes published by Beijing Meteorological Bureau. There are missing and dirty values in these datasets. In order to exploit the data as much as possible, we fill in missing values with the mean in a period of time and delete the tuples with too many consecutive missing data; Table 2 shows the details of these datasets.

5.2. Experiment Setting. There are two groups of experiments. In the first group, we compare GAT-LSTM with the most effective method to date and benchmark models. For each dataset, we divide it into train set and test set, then train these models on the same train set and evaluate their effectiveness on the same test set. The models used for comparison include the following:

(i) *ARIMA: Auto Regressive Integrated Moving Average.* ARIMA is the most common statistical model used for time series forecasting

(ii) *LSTM [32].* LSTM can learn the time dependence in time series. Compared with RNN, they can deal with longer time series and obtain better results

(iii) *ST-DNN [33].* The spatiotemporal models combing of Convolution Neural Networks (CNN) and LSTM for air quality prediction. ST-DNN is the most effective method to date

In the second group, we set one city as target city and the other cities as source cities. Then, delete most of the data from the target city and only keep a small part for training. The proposed metalearning algorithm is used to build a prediction model by transfer learning knowledge from source cities with sufficient data. We compare the proposed metalearning algorithm MetaGAT-LSTM with the following transfer learning methods:

(i) *Fine-Tuning.* First, use the data of a single city to pretrain the GAT-LSTM model and, then, fine-tune the model on the target city, which is called the single-source domain fine-tuning (Single-FT). Secondly, use the data from multiple source cities to pretrain the GAT-LSTM model then fine-tune it on the target city, which is called the multisource domain fine-tuning (Multi-FT).

(ii) *MAML [28].* Use data from all cities to jointly train the model for the target. MAML is implemented based on the metalearning method

The target pollutant is AQI (can be seen as a single pollutant). We use Root Means Square Error (RMSE), Mean Absolute Error (MAE), and ACCuracy (ACC) to evaluate models, which are defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathscr{T}|} \sum_{(\boldsymbol{X},\boldsymbol{y}) \in \mathscr{T}} \|f(\boldsymbol{X}) - \boldsymbol{y}\|_2^2}, \tag{9}$$

$$\mathrm{MAE} = \frac{1}{|\mathscr{T}|} \sum_{(\boldsymbol{X},\boldsymbol{y}) \in \mathscr{T}} \|f(\boldsymbol{X}) - \boldsymbol{y}\|_1, \tag{10}$$

Table 2: Data details.

Attribute		Number of different items			
		Beijing	Tianjin	Shenzhen	Guangzhou
Time span		2014/5/1-2015/4/30	2014/5/1-2015/4/30	2014/5/1-2015/4/30	2014/5/1-2015/4/30
AQIs		278023	189604	88139	281436
Station ID		36	27	11	42
Major pollutants	PM_{10} NO_2 CO O_3 SO_2	278023	189604	88139	281436
Historical weather	District ID	17	20	7	5
	Basic weather Temperature Pressure Humidity Wind speed Wind direction	116867	106614	30305	55632
Weather forecast	District ID	17	20	6	5
	Basic weather Wind strength Wind direction	390702	361624	106380	51870

$$\mathrm{ACC} = 1 - \frac{1}{|\mathscr{T}|} \sum_{(\boldsymbol{X},\boldsymbol{y})\in\mathscr{T}} \frac{\|f(\boldsymbol{X}) - \boldsymbol{y}\|_1}{\|\boldsymbol{y}\|_1}. \tag{11}$$

Here, $\mathscr{T}$ is the test set. $\boldsymbol{y}$ is sample's label (true monitoring data in the future) and $f(\boldsymbol{X})$ is the predicted value of $\boldsymbol{y}$. $\|\bullet\|_1$ and $\|\bullet\|_2$ are L1 and L2 norm, respectively.

In GAT-LSTM, the dimension of the GAT's output vector, the LSTM's output vector, and cell state vector are all set to 128. While training, we use dropout [34] and batch normalization [35] to strengthen the training effect. The batch size is set to 64. The number of epochs is set to 3.

5.3. Experiment Results. At first, we need to find an appropriate influence radius r for building the directed graph G_r in GAT-LSTM, so we compare the performance of GAT-LSTM with different r. Table 3 and Figure 5 give the comparison results on the dataset from Beijing. They show that when $r < 20$km, the accuracy of GAT-LSTM increases as r increases. The reason for this phenomenon is that when r is within a reasonable range, a larger r allows the model to consider more spatial correlation, thereby providing a more accurate prediction. The accuracy reaches its peak at $r = 20$ km. When $r > 20$km, the accuracy decreases slightly as r increases. This phenomenon is because too large r will cause the model to incorrectly estimate the correlation among some remote monitoring sites based on the data similarity. Thus, we set $r = 20$ km in the following experiments.

In the first group of experiments, we use the dataset from Beijing to evaluate all the prediction models. We divide the dataset into training set (70%), validation set (20%), and test set (10%). Each model takes data from the past 48 hours as input ($k + 1 = 48$), then outputs prediction values for the next l hours. Table 4 shows the experiment results with different l. The best results are marked in bold. It can be seen that the traditional linear model ARIMA does not perform well under the influence of multiple complex factors. LSTM's performance is acceptable for short-term prediction and drops quickly with the increase of l. Spatial correlation plays an important role in air quality prediction. By using CNN to extract spatial correlation among monitoring stations, the ST-DNN performs much better than ARIMA and LSTM. However, the spatial correlation built by ST-DNN cannot change dynamically with the change of weather, which reduces its predictive effects. By using GAT to dynamically model spatial correlation, GAT-LSTM gives the best performance in all cases. The performance of all models declines with the increase of l, but the decline rate of GAT-LSTM is lower than the other three, which shows that it is suitable for long-term prediction.

Table 3: The performance of GAT-LSTM with different influence radius r on Beijing dataset ($l = 6$).

r (km)	RMSE	MAE	ACC
5	39.1	24.8	0.767
10	33.3	21.9	0.778
20	28.5	20.1	0.799
25	30.3	20.5	0.797
40	29.7	20.2	0.796
60	28.8	20.5	0.798
100	29.3	20.2	0.796

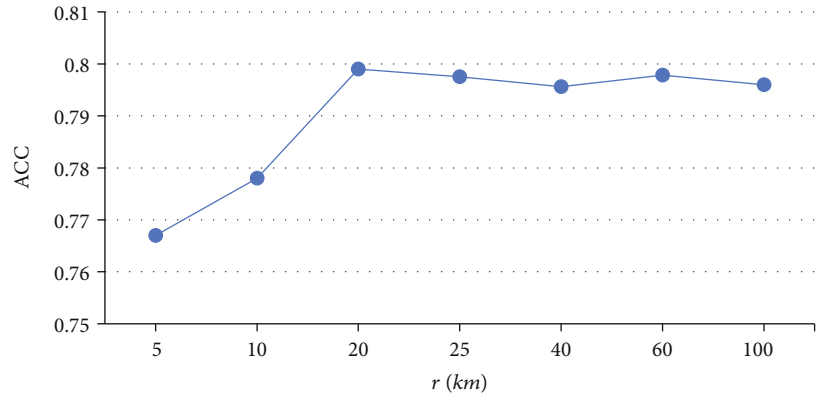


Figure 5: The relationship between influence radius r and prediction accuracy ($l = 6$).

Table 4: Comparison of mean prediction results by different methods among 36 monitoring stations in Beijing (RMSE, MAE, ACC).

Methods	+6 h			+12 h			+24 h			+48 h		
	RMSE	MAE	ACC	RMSE	MAE	ACC	RMSE	MAE	ACC	RMSE	MAE	ACC
ARIMA	53.3	40.7	0.622	70.7	59.2	0.432	94.1	68.2	0.321	104.3	78.2	0.201
LSTM	48.2	34.5	0.723	57.3	47.3	0.582	71.2	55.1	0.473	85.7	66.3	0.367
ST-DNN	35.6	23.8	0.761	52.2	37.1	0.67	63.2	49.3	0.546	70.4	57.7	0.474
GAT-LSTM	28.5	20.1	0.799	47.9	34.8	0.698	55.7	45.2	0.621	65.8	48.9	0.501

Table 5: RMSE of transfer learning methods over different sizes of training dataset in the target city. (taking Beijing as the target city and other cities as the source cities).

Methods		24 h	72 h	240 h	720 h	2400 h
Single-FT	Tianjin	62.7	62.1	53.5	50.5	45.6
	Shenzhen	71.3	70.7	67.6	61.0	55.2
	Guangzhou	70.9	69.0	66.2	59.8	54.5
Multi-FT		63.1	61.2	55.3	52.5	43.1
MAML		59.4	57.8	49.4	45.1	36.2
MetaGAT-LSTM		55.8	53.7	45.3	40.2	31.4

Table 6: RMSE of transfer learning methods over different sizes of training dataset in the target city. (taking Shenzhen as the target city and other cities as the source cities).

Methods		24 h	72 h	240 h	720 h	2400 h
Single-FT	Tianjin	77.3	76.2	72.0	66.3	58.1
	Beijing	79.2	77.2	73.1	68.9	59.2
	Guangzhou	65.6	60.3	51.9	49.3	43.7
Multi-FT		66.3	65.1	58.9	56.7	45.4
MAML		61.5	60.5	51.6	48.2	40.3
MetaGAT-LSTM		58.2	57.5	52.3	47.7	35.6

In the second group of experiments, we execute two experiments by taking Beijing and Shenzhen as the target cities, respectively. We delete most of the data from the target city and only keep a small part for training. With different sizes of the training dataset (in target city), the results of comparing MetaGAT-LSTM with other transfer learning methods are given by Tables 5 and 6. It can be seen all the methods performs better with larger training dataset. Table 5 shows that Single-FT from the Tianjin is better than that from the other two cities. Table 6 shows that Single-FT from Guangzhou is better than that from the other two cities. The climate and geographical location cause similarity

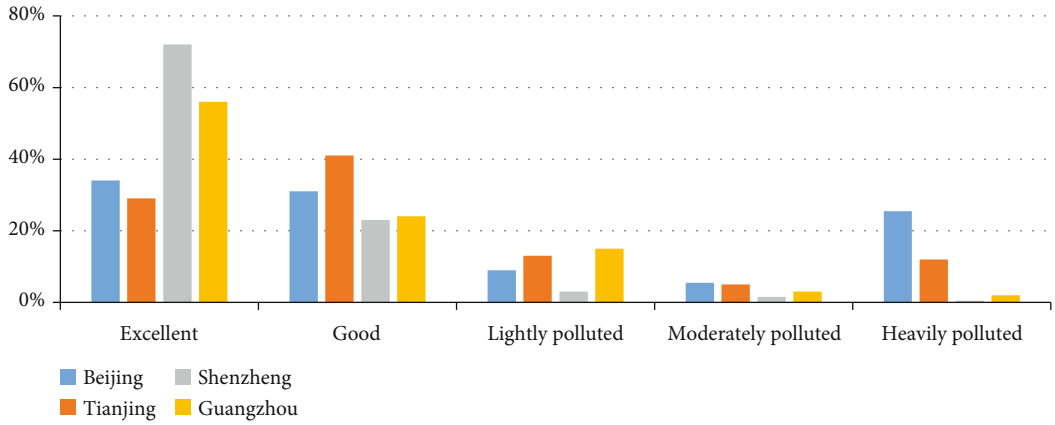


Figure 6: AQI distribution map of four cities.

of air conditions in Tianjin and Beijing, as well as the similarity of air conditions in Guangzhou and Shenzhen. This can be proven by Figure 6, in which the AQI distribution of the four cities from 2014/5/1 to 2015/4/30 is given. The more similar the two cities' datasets are, the better Single-FT performs. Multi-FT enriches the training samples by using the data from all source cities. It is better than Single-FT in some cases. However, because of simply mixing all datasets, it may cause negative migration and give an even worse performance compared with Single-FT in some cases. Both MAML and MetaGAT-LSTM are better than the fine-tuning methods. MetaGAT-LSTM outperforms MAML in all cases by more rationally integrating data from all cities for joint training.

6. Conclusions

In this paper, we propose a spatiotemporal model GAT-LSTM by combining LSTM and GAT for air quality prediction, then design a metalearning algorithm for GAT-LSTM for transfer learning. By more accurately modeling the temporal and spatial correlation of pollutants at all monitoring stations, GAT-LSTM gives a better performance compared with the up-to-date air quality prediction models. In the case of insufficient training data from the target city, the proposed metalearning algorithm for GAT-LSTM can effectively transfer knowledge from source cities with sufficient data and jointly training an accurate prediction model. A number of comparative experiments show the effectiveness of the proposed prediction model and metalearning algorithm. In the future, we may consider more factors related to air quality to improve prediction's accuracy. On the other hand, it will be reasonable to apply the proposed model and metalearning algorithm to other fields.

Data Availability

The source code implemented in this article can be obtained from a GitHub repository (https://github.com/123scarecrow/paperCode), which also includes data analysis code, data pre-processing code, and training data generation code. The data used in the experiments comes from the website: http://research.microsoft.com/apps/pubs/?id=246398.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200, the National Natural Science Foundation of China under Grant No. 62072135 and 61672181, and the Fundamental Research Funds for the Central Universities under Grant No. 3072020CF0602 and 201-510318070.

References

[1] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering (TNSE)*, vol. 7, no. 2, pp. 766–775, 2020.

[2] X. Fang, J. Luo, G. Luo, W. Wu, Z. Cai, and Y. Pan, "Big data transmission in industrial IoT systems with small capacitor supplying energy," *IEEE Transactions on Industrial Informatics (TII)*, vol. 15, no. 4, pp. 2360–2371, 2019.

[3] W. Cheng, Y. Shen, Y. Zhu, and L. A. Huang, "A neural attention model for urban air quality inference: learning the weights of monitoring stations," in *32nd AAAI Conference on artificial intelligence (AAAI 2018*, pp. 2151–2158, New Orleans, USA, 2018.

[4] H. P. Hsieh, S. D. Lin, and Y. Zheng, "Inferring air quality for station location recommendation based on urban big data," in *21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2015)*, pp. 437–446, Sydney, Australia, 2015.

[5] L. D. Mercer, A. A. Szpiro, L. Sheppard et al., "Comparing universal kriging and land-use regression for predicting concentrations of gaseous oxides of nitrogen (NOx) for the Multi-Ethnic Study of Atherosclerosis and Air Pollution (MESA Air)," *Atmospheric Environment*, vol. 45, no. 26, pp. 4412–4420, 2011.

[6] A. Shamsoddini, M. R. Aboodi, and J. Karami, "Tehran air pollutants prediction based on random forest feature selection method," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. XLII-4/W4, pp. 483–488, 2017.

[7] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *Journal of the ACM*, vol. 22, no. 10, p. 111, 2020.

[8] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *31st AAAI Conference on artificial intelligence (AAAI 2017)*, pp. 1655–1661, San Francisco, USA, 2017.

[9] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR 2018)*, Vancouver, Canada, 2018.

[10] Y. Zhang, Q. Lv, D. Gao et al., "Multi-group encoder-decoder networks to fuse heterogeneous data for next-day air quality prediction," in *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pp. 4341–4347, Macao, China, 2019.

[11] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *The 39th IEEE International Conference on Distributed Computing Systems (ICDCS 2019)*, vol. 2019no. 1, pp. 144–153, Dallas, USA, 2019-July.

[12] Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-i.i.d. data in AIoT," *IEEE Transactions on Industrial Informatics*1.

[13] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the heterogeneity: a self-organized federated learning framework for IoT," *IEEE Internet of Things*, vol. 8, no. 5, pp. 3088–3098, 2021.

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[15] M. Ribeiro, K. Grolinger, H. F. ElYamany, W. A. Higashino, and M. A. M. Capretz, "Transfer learning with seasonal and trend adjustment for cross-building energy forecasting," *Energy and Buildings*, vol. 165, pp. 352–363, 2018.

[16] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Transfer learning for time series classification," in *IEEE International Conference on Big Data (Big Data 2018)*, pp. 1367–1376, Seattle, USA, 2018.

[17] P. Xiong, Y. Zhu, Z. Sun et al., "Application of transfer learning in continuous time series for anomaly detection in commercial aircraft flight data," in *In IEEE International Conference on Smart Cloud (Smart Cloud)*, pp. 13–18, 2018.

[18] M. Lippi, M. Bertini, P. Frasconi et al., "Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] Z. Luo, J. Huang, K. Hu, X. Li, and P. Zhang, "Accu air: winning solution to air quality prediction for KDD Cup 2018," in *25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2019)*, pp. 1842–1850, Anchorage, USA, 2019.

[21] Y. Zheng, F. Liu, and H. P. Hsieh, "U-air: when urban air quality inference meets big data," in *19th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2013)*, pp. 1436–1444, Chicago, USA, 2013.

[22] C. J. Huang and P. H. Kuo, "A deep cnn-lstm model for particulate matter (PM2.5) forecasting in smart cities," *Sensors*, vol. 18, no. 7, p. 2220, 2018.

[23] J. Bruna, W. Zaremba, A. Szlam, and Y. Le Cun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR 2014)*, Banff, Canada, 2014.

[24] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3844–3852, 2016.

[25] C. Zhang, J. Q. James, and Y. Liu, "Spatial-temporal graph attention networks: a deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166246–166256, 2019.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR 2015)*, San Diego, USA, 2015.

[27] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83–95, 2016.

[28] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, vol. 70, pp. 1126–1135, 2017.

[29] J. Schmidhuber, *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-hook*, Technische Universität München, New York, NY, 1987.

[30] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: a meta-learning approach for spatial-temporal prediction," in *The World Wide Web Conference*, pp. 2181–2191, 2019.

[31] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *In Advances in neural information processing systems*, pp. 5998–6008, 2017.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] P. W. Soh, J. W. Chang, and J. W. Huang, "Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations," *IEEE Access*, vol. 6, pp. 38186–38199, 2018.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 34th International Conference on Machine Learning (ICML 2015)*, vol. 37, pp. 448–456, Lille, France, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5536386, 14 pages
https://doi.org/10.1155/2021/5536386

Research Article

Blind Travel Prediction Based on Obstacle Avoidance in Indoor Scene

Zhiqiang Lv, Jianbo Li, Haoran Li, Zhihao Xu, and Yue Wang

College of Computer Science & Technology, Qingdao University, 266071, China

Correspondence should be addressed to Jianbo Li; lijianbo@qdu.edu.cn

Received 17 February 2021; Accepted 6 June 2021; Published 23 June 2021

Academic Editor: Xiao Zhang

Blind people have intelligent tools to rely on for travel with the development of navigation technology. The GPS navigation, blind track, etc., are tools that blind people often use when traveling outdoors. However, indoor navigation tools and technology for blind people are lacking. We propose an obstacle avoidance algorithm and a spatial-temporal model of trajectory prediction for the indoor travel task of the blind. The focus of this work is that it enables the blind to accurately avoid obstacles and achieve high accuracy trajectory prediction aiming at the unique movement characteristics of the blind. We set up a variety of baselines to conduct an experimental evaluation on a dataset of blind trajectories in a multistorey shopping mall. The experimental results show the advantages of the data model and predictive model of this work.

1. Introduction

With the rapid development of the transportation and automobile industry, the trajectory prediction has become a focus in the field of transportation big data. It helps people plan better travel routes as well as saving more manpower and material resources. More importantly, the public basic resources are allocated more precisely with the help of trajectory prediction. Blind people live in a dark world, which brings great difficulties to work, life, and social activities [1]. How to walk safely is the biggest problem in the life of the blind. At present, there are many researchers in the world who solve the problem of outdoor navigation for the blind, such as voice broadcast systems in public places (blind roads and bus stops) and GPS (Global Positioning System) for outdoor navigation for the blind. However, there are few results in indoor navigation for the blind. The GPS navigation functions are already very mature. However, GPS is mainly used for outdoor travel activities for blind people. Even if they go out, they usually go to hospitals, hotels, and other indoor buildings. So, the indoor scenes are the main activity areas for blind people and they have their own different channels and obstacle spatial distribution. Moving indoors without GPS is a problem for blind people. Therefore, the research of indoor navigation projects has extremely important social significance and research value. There are two important considerations in indoor navigation for the blind, namely, the accurate prediction of the trajectory of the blind and the accurate avoidance of indoor static objects.

Blind people need auxiliary equipment to issue motion instructions for them during walking, so the trajectory prediction for blind people is an important basis for generating motion instructions. Pedestrian trajectory prediction models are mainly divided into traditional mathematical-statistical models and data-driven neural network models [2]. Traditional mathematical-statistical models rely on artificially designed features to model pedestrian actions and goals. The Social Force Model (SFM) proposed by Helbing and Molnar [3] transforms the interaction between pedestrians and pedestrian goals into gravitation and repulsion. This work believes that the goal of pedestrians can attract pedestrians to the goal. The repulsion among pedestrians prevents pedestrian collision. Trautman and Krause [4] improve the SFM with an interactive Gaussian process. They use the Gaussian process to predict the trajectory of each pedestrian and calculate the probability of the prediction result according to the potential function of SFM. The Markov model can make probabilistic spatial-temporal prediction of pedestrian

trajectory [5] [6]. The training process of the model can dynamically adjust the training parameters with the help of reinforcement learning [7]. It can make the prediction process consider the physical influence of the outside world and make the predicted trajectory closer to the actual trajectory. The above methods have the advantages of simplicity, intuitiveness, and low complexity, but their process of building a model is too sensitive to calculate parameters, and the generalization ability of the model is weak. More importantly, the above methods can only simulate the short-term reaction of pedestrians and cannot consider the long-term historical information of the location.

In data-driven forecasting tasks, the recurrent neural network (RNN) has obvious advantages over traditional mathematical-statistical models [8], especially in the long-term and time-dependent feature calculation process. The RNN is a neural network used to process sequence data. Compared with the general neural network, it can process the data of the sequence change. The Long Short-Term Memory (LSTM) [9] is a special RNN that can solve the problems of gradient disappearance and gradient explosion in the training process of long sequences. More importantly, the LSTM can perform better in longer sequences than ordinary RNNs. The LSTM can not only realize the sequence prediction of pedestrian position but also calculate the mutual influence among different pedestrians [10]. However, the LSTM has the disadvantage that the RNN cannot capture the high-level spatial-temporal structure [11]. In order to overcome this disadvantage and maintain the characteristics of pedestrian trajectory, Alahi et al. [12] propose the Social Long Short-Term Memory (S-LSTM) model. The S-LSTM collects the hidden state of adjacent pedestrians by introducing a social pooling layer and shares the hidden information of adjacent pedestrians by the spatial distance of a grid. In order to reduce information loss, Vemula et al. [13] replaced the social pooling layer with the social attention layer. The social attention layer forms interactive features by assigning weights among pedestrians in the grid. Unlike the walking process of normal people, the walking speed of blind people is very slow, because they can make the next decision to walk only after they have fully explored the current road. The above-mentioned research works all focus on predicting the trajectory of normal people, so they lack feature calculations for the movement characteristics of blind people.

The trajectory prediction of a blind person realizes the prediction of the short-term future position of the blind based on the characteristics of motor behavior. Any building should be regarded as an obstacle for the blind. Blind people should avoid obstacles fully when walking [14]. The above movement characteristics of the blind determine the focus of this work on how to achieve accurate prediction of the position of the blind and how to avoid obstacles. The movement characteristics of the blind determine the focus of this work on how to realize the accurate prediction of the position of the blind while avoiding obstacles. Wang et al. [15] propose a path planning algorithm for blind navigation systems. It uses the Dijkstra algorithm [16] as the basic algorithm and relational database as the storage mode. The algorithm uses a multifactor fuzzy algorithm to calculate the weight of obstacles in the road network. Its core is to make an adjacency matrix according to the spatial distribution of obstacles in the road network. Finally, it makes the topological structure diagram of the obstacle network. However, this obstacle network is a method of representing the position of obstacles in a local area. When the position of the blind person changes, the obstacle network needs to be recalculated. This design mode has high computational complexity and cannot consider the spatial distribution of obstacles on a global scale. The Graph Convolutional Network (GCN) [17] shows superiority in the representation of global spatial relations. The theory of convolutional neural network (CNN) [18] is to use random and shared convolution kernel to get the weighted sum of pixels. Then, it uses backpropagation to optimize the convolution kernel parameters to automatically extract features. However, many data in real life are stored in the form of graphs, such as social network information, knowledge graphs, protein networks, and the World Wide Web. The form of these graph networks is not like the images that are neatly arranged in matrix form but unstructured data. The GCN has a general paradigm for calculating graph features. More importantly, the GCN can use the adjacency matrix representing the connectivity of nodes to represent the connectivity among the entire spatial location [19]. This model can be used to calculate the spatial distribution of obstacles in indoor spaces. For the blind, the blind should avoid obstacles, which means that the location of the obstacle does not have connectivity.

In response to the above research, we propose a Blind Trajectory Prediction Model (BlindTPM). The BlindTPM is used for the tasks of blind trajectory prediction and obstacle avoidance in indoor scenes. The main contributions of this work are as follows:

(i) We propose a spatial-temporal model for pedestrian trajectory prediction. The spatial convolution block captures the spatial relative characteristics of roads and obstacles. Different from the existing pedestrian trajectory model, this model combines the unique position change characteristics of the blind to realize the trajectory prediction task. According to the slow motion of blind people, the temporal convolution block designed by this model can capture long-term historical information of pedestrian trajectories

(ii) We propose a trajectory algorithm for obstacle avoidance. The algorithm calculates the spatial distribution and connectivity of obstacles with the help of the GCN and connectivity adjacency matrix

(iii) We design three data-driven methods to verify the performance of BlindTPM, including a trajectory prediction method based on abscissa and ordinate, a trajectory prediction method based on a grid map, and a trajectory prediction method based on a grid map with obstacle distribution. We model the trajectory data of blind people in a multifloor shopping mall. Compared with the existing

baselines, the BlindTPM has obvious performance advantages

2. Data Design

We design three data-driven methods to complete the task of blind trajectory prediction, including a trajectory prediction method based on abscissa and ordinate, a trajectory prediction method based on a grid map, and a trajectory prediction method based on a grid map with obstacle distribution.

2.1. Abscissa and Ordinate. The abscissa and ordinate are used to represent the position of the blind in the design idea of this method, as shown in Figure 1. The $X = \{x_1, x_2, x_3, \cdots, x_t\}$ and $Y = \{y_1, y_2, y_3, \cdots, y_t\}$. Because this work is aimed at a fixed blind moving scene indoors, the X and Y are relative positions to a fixed origin, not latitude and longitude. The t is the time step of the blind position change. This method implements the most basic trajectory prediction task, which uses t consecutive positions to predict $t + n$ positions in the future, as shown in

$$\left(X', Y'\right) = \sigma'(\mathrm{BPT}\langle\sigma(X, Y)\rangle). \tag{1}$$

The $X^{'} = \{x_{t+1}, x_{t+2}, \cdots, x_{t+n}\}$ and $Y^{'} = \{y_{t+1}, y_{t+2}, \cdots, y_{t+n}\}$. The BPT is the modeling method in this work. The σ and σ' represent the data standardization and destandardization process, respectively. Different evaluation indicators usually have different dimensions and units in the field of machine learning. In order to eliminate the dimensional influence between indicators, we need to standardize the data so that the data indicators can be compared. Because the abscissa and ordinate data representing the position of the blind are floating-point numbers and the data changes are limited to a small range, so we use the Z-score as the method of data standardization, as shown in Formula (2). The μ represents the mean of the overall data, and the δ represents the standard deviation of the overall data. The calculation process of Z-score is simple, and it eliminates the impact of data magnitude.

$$Z = \frac{(X \mid Y) - \mu}{\delta}. \tag{2}$$

2.2. Grid Map. There is a very obvious difference between the walking trajectory characteristics of a normal person and a blind person. Figure 2 shows the changes of abscissa and ordinate of two trajectories in the same scene according to time steps. The time step interval of the two trajectories is the same, which is two seconds. The trajectory of a normal person is basically a linear change, which has obvious characteristics of change. However, the position of the blind does not change frequently due to the constant movement and the need to explore the road. This situation is reflected by the continuous same value of the abscissa and ordinate of the trajectory in Figure 2. Finally, it causes a loss of accuracy due to a large number of sample repetitions in the weight calculation process of weight of actual trajectory.

In order to deal with the above situation, we use the local sampling [20] method to clean the dataset of the blind trajectory, as shown in Figure 3. We use sample points to replace multiple locally unchanged points in the abscissa and ordinate. The data shows obvious trajectory features after being cleaned, as shown in Figure 4(a). We can know that the position of a blind person can be represented by the label of a grid when the indoor scene is mapped to a grid. This method transforms the regression prediction task of this work into a classification task when the indoor scene of the blind movement is relatively single and the number of divided grids is large.

The calculation method of dividing the track points into grid labels is shown in Formula (3). The $\propto$ is the coordinate dimension represented by a grid label, which is equal to 2 in this work. Formula (3) is applicable to the grid label calculation process of abscissa and ordinate. More importantly, it takes into account the positive and negative values of coordinate data.

$$l_i^{(x)} = \begin{cases} \left\lceil \dfrac{\max(X) - x_i + 1}{\propto} \right\rceil, & x_i > 0, \\ \left\lceil \dfrac{|\min(X)| + x_i + 1}{\propto} \right\rceil, & x_i < 0. \end{cases} \tag{3}$$

According to the above method, we can get the grid labels of abscissa ($l_i^{(x)}$) and ordinate ($l_i^{(y)}$). We expect to obtain n positions of the blind in the future according to the historical trajectory, so the $L^{(x)}$ and $L^{(y)}$ in Formula (4) are sequence data. The $L^{(x)} = \{l_{i+1}^{(x)}, l_{i+2}^{(x)}, \cdots, l_{i+n}^{(x)}\}$ and $L^{(y)} = \{l_{i+1}^{(y)}, l_{i+2}^{(y)}, \cdots, l_{i+n}^{(y)}\}$. Generally speaking, a location point is described by its abscissa and ordinate in the 2-dimensional plane. So, we take this into consideration when evaluating the accuracy of the model. Only when the predicted values of the grid labels of the abscissa and ordinate are the same as the true values in the same time step can it be considered as a successful prediction.

$$\left(L^{(x)}, L^{(y)}\right) = \mathrm{LogSoftmax}(\mathrm{BPT}\langle X, Y\rangle). \tag{4}$$

2.3. Obstacle Distribution. The spatial distribution of obstacles is an important factor affecting the walking process of blind people indoors. Figure 5(a) shows the spatial distribution of obstacles (buildings, decorations, etc.) in the actual indoor scene. Among them, the black areas represent obstacles. Figure 5(b) is a standardized obstacle distribution map, which is derived from the data cleaning of (a). We must admit that (b) there is a certain information error, so we try our best to increase the number of obstacles to avoid the problem of missing obstacles. Finally, the grid area where obstacles exist is assigned the value 0 and the other grid areas are assigned the value 1, as shown in Figure 5(c).

We design an obstacle avoidance algorithm for blind people based on the connectivity among grids, as shown in Figure 6. The main theory of the algorithm is to use the connectivity among grids to construct an adjacency matrix to

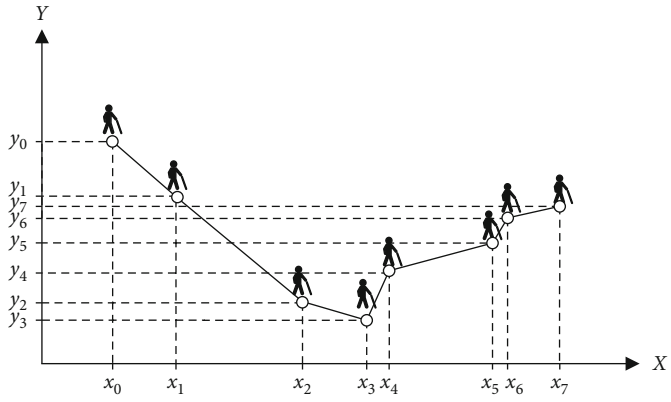


Figure 1: Basic track position representation. The location point changes according to time. The X and Y represent the abscissa and ordinate of the position, respectively.

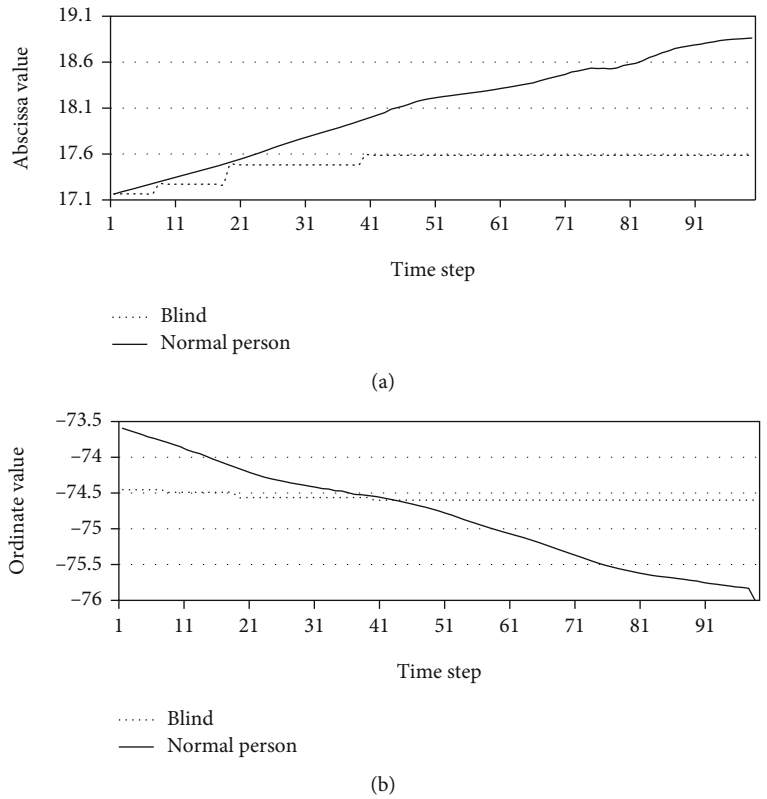


Figure 2: The difference between the trajectory characteristics of the blind and normal person: (a) the change curve of the abscissa of trajectory with time and (b) the change curve of the ordinate of trajectory with time.

capture the global spatial distribution characteristics of roads and obstacles in indoor scenes. The adjacency matrix is a data structure used to describe the relationship between vertices and edges. It is essentially a two-dimensional array and is suitable for dealing with the relationship among the smallest data units. Blind people can walk in up to nine directions in a grid, including front, back, left, right, front left, front right, back left, back right, and motionless. These nine directions correspond to the eight adjacent grids of a grid and the grid where it is located. However, the presence of obstacles (the black grid in Figure 6(a)) makes the connectivity of a grid to nine directions uncertain. According to the obstacle spatial distribution grid in Figure 5(c) and grid connectivity rule, we design an algorithm for calculating grid connectivity, which is suitable for the computing process of the connectivity adjacency matrix among grids with obstacles, as shown in Algorithm 1. The output A of Algorithm 1 is a symmetric adjacency matrix, as shown in Figure 6(b). Its symmetry is shown in that the connectivity between one grid and another grid is equivalent in the forward and reverse interaction process. If there are obstacles in a grid, the grid is defined as that it cannot interact with any other grids and 9 directions are assigned the value 0.

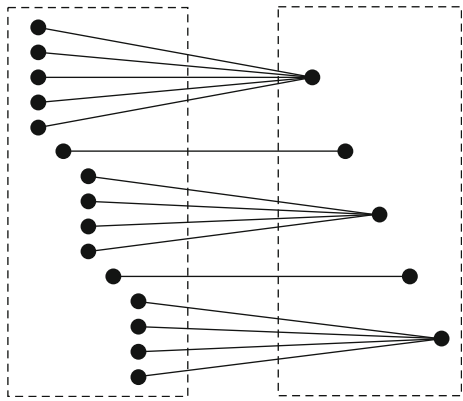


Figure 3: Sample trajectory design. The left side is the original trajectory, and the right side is the sample trajectory.

3. Model Design

We design a deep spatial-temporal model (BlindTPM) that can train, evaluate, and predict the trajectory data of blind people. The BlindTPM consists of three blocks, including spatial convolution block, temporal convolution block, and estimation block, as shown in Figure 7. The spatial convolution block is mainly used to calculate the spatial distribution of trajectories and obstacles. The temporal convolution block is mainly used to calculate the time recursive characteristics of trajectory data. The estimation block is mainly used to reduce the global error and local error of the trajectory prediction result.

3.1. Spatial Convolution Block. The walking process of the blind is complex and diverse. Although the prediction of straight lines is simple, it is often the case that the traveling route becomes a complex curve due to the influence of turns and obstacle distribution. This makes it particularly important to capture the spatial relationship of positions in the feature extraction process of trajectory prediction. Therefore, it is necessary to perform another round of extraction of feature from the trajectory and strengthen the weight of the local curve of the trajectory before making temporal predictions, instead of using standardized position data directly [21]. First of all, the data of abscissa and ordinate are fused to calculate the feature of spatial dependencies, as shown in

$$p_i = \sigma\langle \mathrm{cat}(\delta^{x_k} \times x_i + \mu^{x_k}, \delta^{y_k} \times y_i + \mu^{y_k})\rangle. \tag{5}$$

The abscissa and ordinate of the ith point are calculated with the standard deviation δ and mean μ of the kth trajectory sequence, which represents standardization of position data. The result of cat that fuses the abscissa and ordinate into the same dimension is closer to the real result compared to the result of processing abscissa or ordinate matrix separately. The correlation between the spatial dependence strength of the nodes of the blind trajectory and the contextual information of the trajectory feature is very important. Traditional convolutional networks usually use downsampling to perform high-dimensional feature calculations. However, the fixed receptive field of the downsampling method will cause the loss of edge feature information. Particularly for the trajectory of the blind, the walking process of the blind is very slow, which leads to a lot of trajectory nodes in a certain period of time. The above process makes the weight of the trajectory involve more parameters, and the parameter information becomes very easy to lose. Therefore, we build multilayers of dilated convolution to achieve the process of capturing the complex spatial dependence of trajectory of the blind, as shown in Figure 8. Filter kernels are depicted in a square with a grid pattern in each layer. The black cells in dilated kernel represent valid weights. The dilation factor of the three-layer dilation convolutional network is changed by 2^2, 2^1, and 2^0. The biggest advantage of a dilation convolutional network is that it can expand the receptive field exponentially without losing feature information. The $\{d\}$ is the distance features of the first and the nonfirst nodes of every trajectory.

We combine the o^{conv} and the data which are the distance of the ith node and the first node of the kth trajectory after the activation of the σ (ReLU function) in order to reduce the mutual dependence of parameters and alleviate the problem of gradient disappearance due to overfitting, as shown in

$$O = \sigma\left\langle \mathrm{cat}\left(o^{\mathrm{conv}}, \delta^{d_k} \times d_i + \mu^{d_k}\right)\right\rangle. \tag{6}$$

3.2. Obstacle Grid Spatial Distribution. In order to add the obstacle grid to the modeling process, we use the GCN to fuse the connectivity adjacency matrix with the feature data. The essence of the process of the GCN is that each node of graph is changing its state all the time due to the influence of neighbors and further nodes. The more closely related other nodes have a greater impact on the original node. The method of Laplace can transfer the strength of features in the GCN in proportion to the state difference among them. In order to add the influence of the original node on itself into the calculation process, we use an improved version of the method of Laplace, as shown in

$$\mathrm{La} = O^{-1/2} A O^{-1/2}. \tag{7}$$

The A represents a connectivity adjacency matrix with a self-connected state. We take the output of the spatial convolution block as the degree distribution of the nodes (O). The above formula introduces its own degree matrix to solve the problem of self-transmission and realizes the normalization operation of the adjacency matrix by multiplying the two sides of the adjacency matrix by the degree root of the node and taking the inverse. The original spectrogram convolution implements the filter of the product of each node and the Fourier transform. However, the eigenvectors are high order and the eigendecomposition of the Laplacian matrix is very inefficient in the decomposition process of large graph structures [22]. So, we use the K-order Chebyshev polynomials to approximate the optimization of the Laplacian matrix, as shown in Formula (8). The calculation process of $T_i(\widetilde{\mathrm{La}})$ is shown in Formula (9), which represents the recursive definition of the Chebyshev polynomial. This method is called the K-localized [23] convolution algorithm, which ensures that

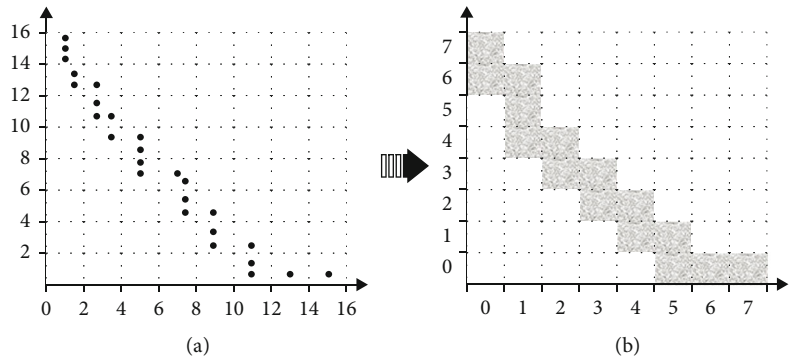


Figure 4: Conversion between blind trajectory and grid label: (a) the distribution of the original trajectory in the map and (b) the grid label distribution of the trajectory.

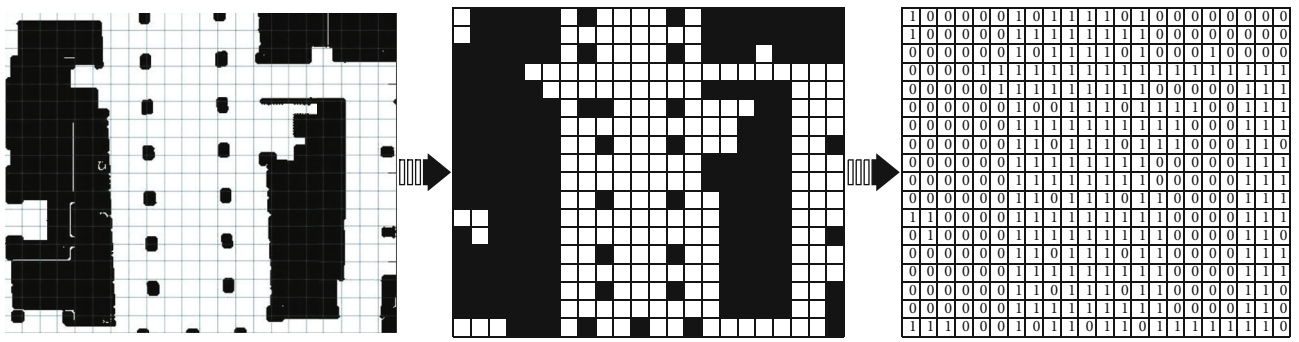
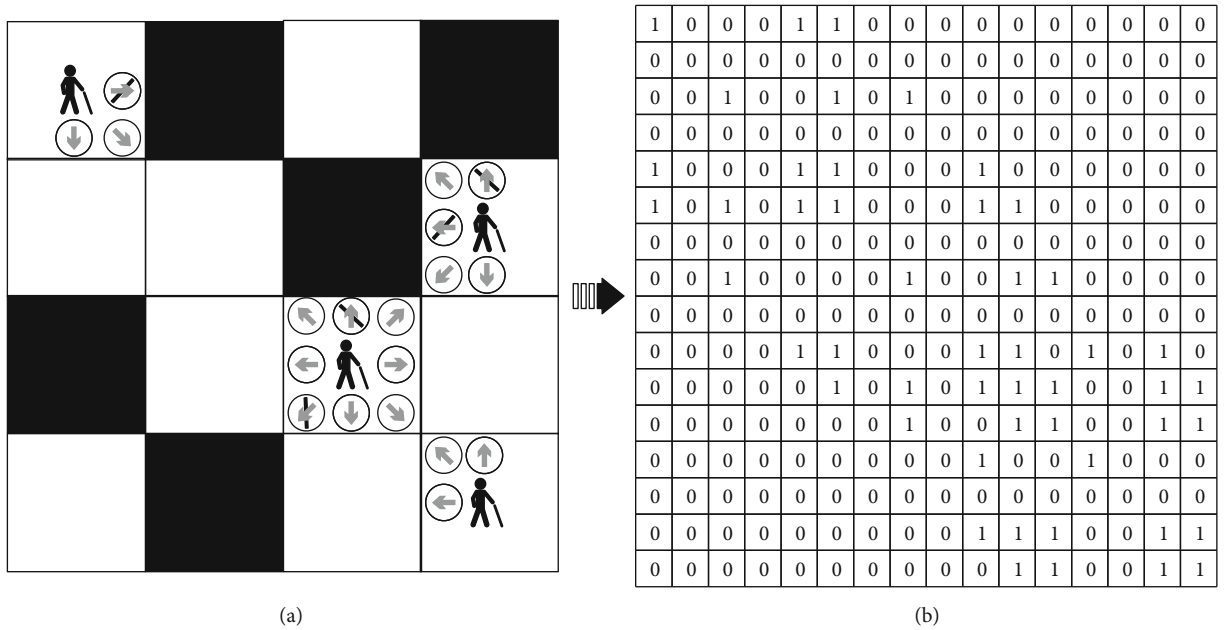


Figure 5: Matrix representing the spatial distribution of obstacles: (a) the distribution of obstacles in the actual indoor scene, (b) the spatial distribution grid of obstacles after desensitization, and (c) a matrix representing the spatial distribution of obstacles.



(a)

1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
1	0	1	0	1	1	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	1	1	0	1	0	1	0
0	0	0	0	0	1	0	1	0	1	1	1	0	0	1	1
0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1
0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1
0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1

(b)

Figure 6: Grid connectivity adjacency matrix design: (a) the connectivity between obstacle grids and (b) the adjacency matrix of grid connectivity.

Input: The set of obstacle spatial distribution, L; the side length of gird, N.
Output: The adjacency matrix of grid connectivity, A.

```
Initialize a matrix A with 0, the shape of A is (N^2 × N^2).
for each i ∈ [0, N^2 − 1] do
  for each j ∈ [i, N^2 − 1] do
    // Determine whether the current grid is an obstacle grid
    if L_i == 0 then A_ij = 0
    else
      if i == j then A_ij = 1
      else if L_j == 1 then
        // Determine the connectivity of front.
        if j == i − N then A_ij, A_ji = 1, 1
        // Determine the connectivity of left.
        else if j == i − 1 & i%N! == 0 then A_ij, A_ji = 1, 1
        else if j == i + 1 then
          if i < N − 1 then A_ij, A_ji = 1, 1
          end if
          // Determine the connectivity of right.
          if i%N! = 2 & i > N − 1 then A_ij, A_ji = 1, 1
          end if
        // Determine the connectivity of front left.
        else if j == i − N − 1 & i%N! == 0 then A_ij, A_ji = 1, 1
        // Determine the connectivity of front right.
        else if j == i − N + 1 & i%N! = 2 & i > N − 1 then A_ij, A_ji = 1, 1
        // Determine the connectivity of back.
        else if j == i + N then A_ij, A_ji = 1, 1
        // Determine the connectivity of back left.
        else if j == i + N − 1 & i%N! == 0 then A_ij, A_ji = 1, 1
        // Determine the connectivity of back right.
        else if j == i + N + 1 & i%N! = 2 & i > N − 1 then A_ij, A_ji = 1, 1
        end if
      end if
    end if
  end for
end for
final
return A
```

Algorithm 1: A computing algorithm of adjacency matrix of grid connectivity.

the current node only considers the impact of nodes in the K range on it and greatly reduces the time complexity.

$$g(\theta)\bullet o \approx \sum_{i=1}^{k} \theta_i \bullet T_i\left(\widetilde{\mathrm{La}}\right)\bullet o, \tag{8}$$

$$T_i\left(\widetilde{\mathrm{La}}\right) = \begin{cases} 2\widetilde{\mathrm{La}}\bullet T_{i-1}\left(\widetilde{\mathrm{La}}\right) - T_{i-2}\left(\widetilde{\mathrm{La}}\right), \\ T_0\left(\widetilde{\mathrm{La}}\right) = 1, \\ T_1\left(\widetilde{\mathrm{La}}\right) = \widetilde{\mathrm{La}}, \\ \widetilde{\mathrm{La}} = \dfrac{2}{\lambda_{\max}} \times \mathrm{La} - I_n. \end{cases} \tag{9}$$

3.3. Temporal Convolution Block. Recurrent neural networks have played a huge role in the field of time series data prediction. However, traditional recurrent neural networks (RNN, LSTM, GRU, etc.) only involve a single-step calculation method. Single-step calculation has two disadvantages, namely, the complexity of the calculation process is very high and the performance of long-term prediction is low. The Temporal Convolution Network (TCN) [24] is designed based on the idea of convolutional neural network and parallelization. It overcomes the two inherent shortcomings of traditional recurrent neural networks. For blind person trajectory prediction, the TCN uses causal convolution to make all historical location points of the trajectory be associated with predicted future location points. The using of the dilated convolution enables the hidden layer to obtain a larger receptive field to establish the high-dimensional timing features of the blind trajectory. We build the temporal convolution block with 7 hidden layers to calculate the temporal features [21], as shown in Figure 9. Every hidden layer has one dilation factor and every factor is exponential growth

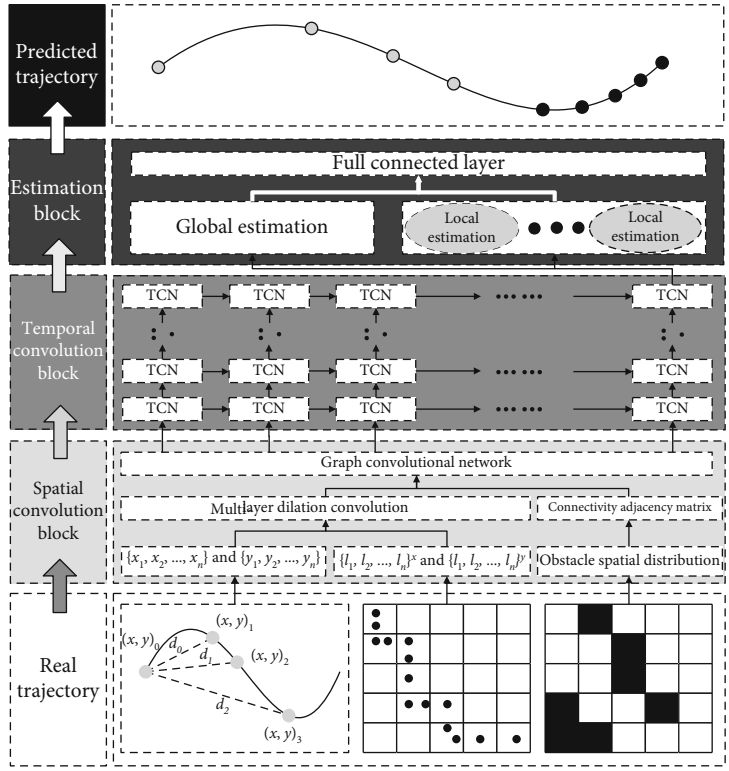


Figure 7: Model architecture design.

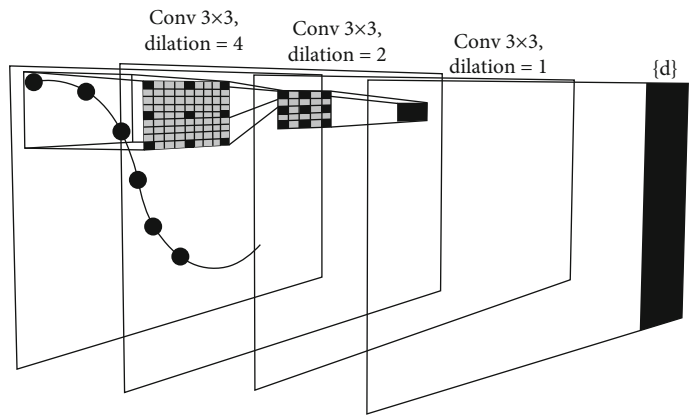


Figure 8: Dilation convolution process.

for 2. The temporal prediction sequence $\{h_0, h_1, \cdots, h_k\}$ is the output of the TCN.

3.4. Estimation Block. In order to improve the accuracy of blind trajectory prediction, we use a local-global estimation [25] block. The purpose of using the local estimation method is to add the information of the next location point for the blind current location point to form a contextual multiscale calculation process. Therefore, each position point of the blind trajectory will go through the process of local estimation [21]. Compared with the trajectory of a normal person, the position of the blind trajectory is denser per unit time. When multiple local estimations are performed, their cumulative errors have a certain degree of influence on the final prediction results. Therefore, we use global estimation to reduce the cumulative error of local estimation and improve the global prediction accuracy.

3.5. Local Estimation. After the modeling process of the spatial convolution block and the temporal convolution block, a multidimensional matrix containing temporal and spatial features is formed. In the process of local estimation, the initial 128 dimensions of each h_k are linearly transformed into 1 dimension. An activation function should be performed after each linear transformation. The purpose is to reduce the noise position points in the blind heart trajectory. We take the Leaky ReLU function that has the advantage of a negative saturation region to make the data more inclined to be saturated in the negative region rather than completely return to zero, as shown in Figure 10.

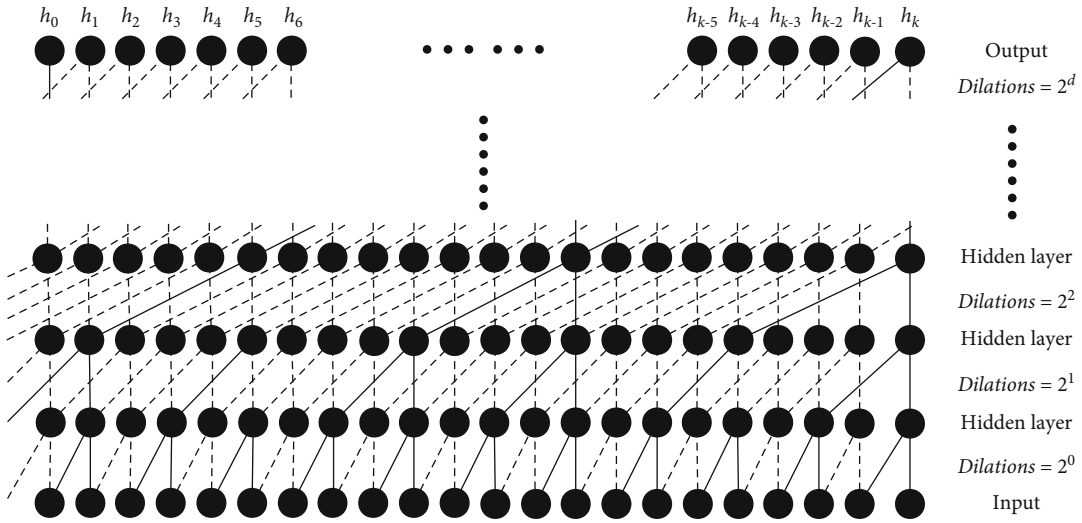


Figure 9: Temporal convolution block design.

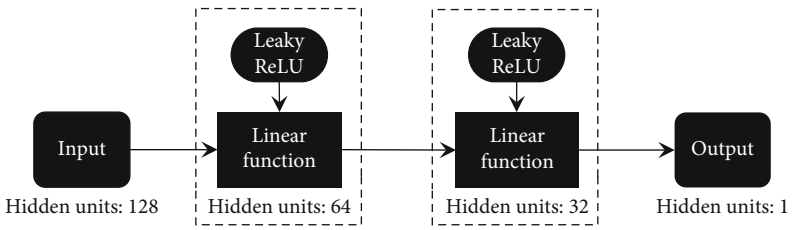


Figure 10: Local estimation design.

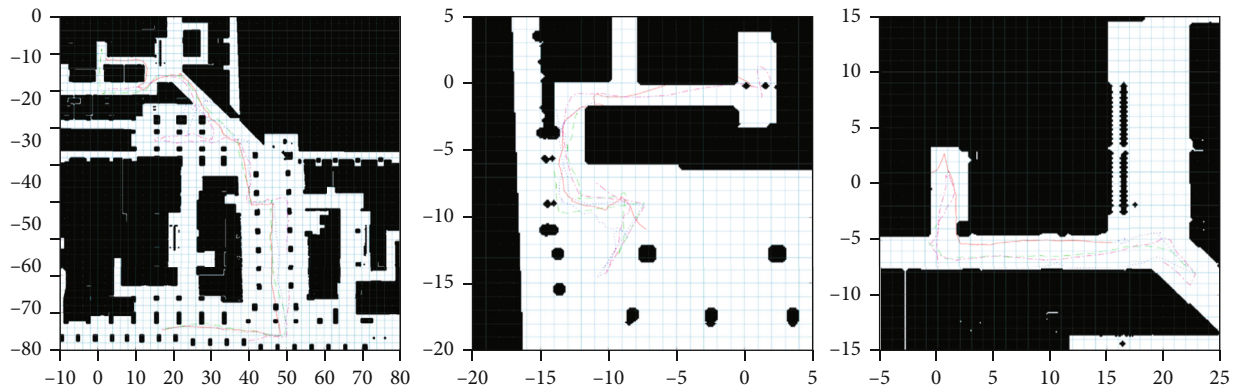


Figure 11: The distribution of the trajectory of the blind in the multistorey shopping mall.

3.6. Global Estimation. We build a three-layer residual network to reduce the cumulative error of local estimation [21]. We fuse the distance features of the first and the nonfirst nodes of every trajectory with the weights. The feature fusion process of each location point will be realized in a three-layer residual network by layer jump connection. The formula is as

$$h_i = \sigma\langle h_{i-1} + \mathrm{re}(i)\rangle. \tag{10}$$

The $\mathrm{re}(i)$ is the ith residual process, and the h_{i-1} is the result of the previous residual process. The input of the residual unit is directly combined with the output of the residual unit. After that, we use the full connection layer to calculate the weight of local estimation and residual network features. The global estimation calculates the weight of a specified number of nodes for each blind trajectory that we need to predict.

4. Experiment

4.1. Data Preparation. The dataset is collected in a multistorey shopping mall with three routes in total [26]. It provides time-stamped collections and annotated trajectories, as well as related floor plans. The following are the trajectories of 9 participants in the three routes. From Figure 11, we can know that the route range of the blind does not

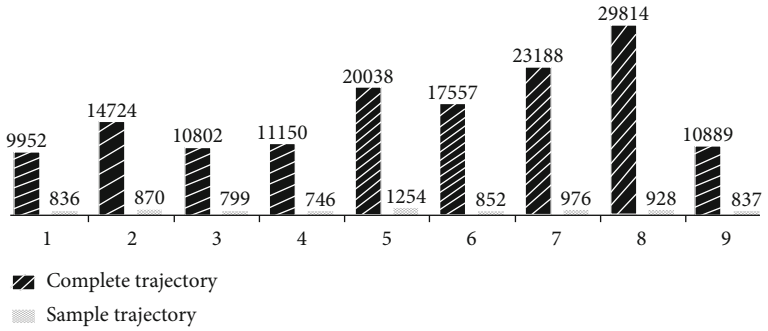


Figure 12: The number of position points contained in the original trajectory and sample trajectory of the blind.

include all routes in the three scenes. Therefore, we select part of each scene for experimental modeling. The area we selected contains the entire track range of the 9 participants.

As mentioned earlier, we sample the original data based on the slow trajectory speed of the blind and their insignificant position change per unit time. The sample trajectory represents the long-term timing characteristics of the trajectory. Figure 12 shows the relationship of position node between the complete and sample trajectories of the 9 participants. The time step is 2~3 s in the complete trajectory, but 100~1000 s in the sample trajectory.

4.2. Performance Evaluation

4.2.1. Baselines. In order to verify the performance of the model under the premise of ensuring the scientificity of the results, we use traditional mathematical-statistical models [3–6], traditional convolutional networks, traditional recurrent networks, multilayer dilation convolutional networks, TCN, and complex spatial-temporal models to conduct comparative experiments. The details of the complex spatial-temporal models are shown as follows:

(i) STF-RNN [27]. It uses a look-up table layer to capture the mixed features of the trajectory in space and time. It inputs this feature into the RNN in an appropriate internal representation method for recursive derivation

(ii) Social-LSTM [12]. It designs a "social" pooling structure to share the parameter hidden state of the end sequence of the LSTMs. The advantage of this design is that the model can automatically learn the interactions that occur among time-coinciding trajectories

(iii) Social Attention [13]. It uses a special Structural RNN (S-RNN) to calculate the weight of the spatial-temporal graph data. It takes the problem content as the node and the time series data as the edge value

(iv) DSCMP [28]. It designs a queue mechanism to explicitly memorize and learn the correlation among long trajectories. It focuses on and uses the consistency characteristics of space and time to capture the contextual parameters of the motion scene

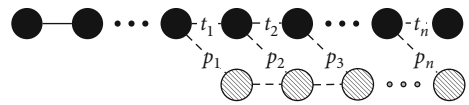


Figure 13: Distance error calculation method.

4.2.2. Coordinate. The training in the experiment uses 4 NVIDIA Tesla V100, and the experimental results are the average value after 200 epochs of training under the premise of consistent datasets. The optimization algorithm selects the Adam algorithm, because the learning rate of each iteration has a certain range after bias correction of the Adam algorithm, which makes the parameters relatively stable. We design the corresponding method of error calculation considering the particularity of the dataset, as shown in Figure 13. The $\{d_1, d_2, \cdots, d_n\}$ is the sequence of distance which is from real points, and $\{p_1, p_2, \cdots p_n\}$ is the sequence of distance which is between nth predicted points and $(n-1)$th real points. We calculate the error of distance of these two sequences.

The Root Mean Square Error (RMSE) can measure the deviation between the predicted value and true value. It is usually used as an indicator to measure the prediction accuracy of a deep learning model. The calculation process of RMSE is as shown in Formula (11). We can know that the BlindTPM performs best in the prediction process of medium and long term from Table 1. Although the Markov has the smallest prediction error for the first point, it loses advantage in the prediction process of medium and long term.

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - y_i\wedge)^2}. \tag{11}$$

4.2.3. Analysis and Expansion. The mathematical-statistical models are aimed at short-term pedestrian actions and goals. The complexity of Markov is low, and its prediction error for the first point in the future is small. However, its error increases linearly compared with other data-driven models as time goes by. The LSTM is a traditional recurrent network, and it can only capture simple time recursive features of location points without other auxiliary data modeling processes. The D-Conv represents a multilayered dilation convolution model. The translation process of the convolution kernel

Table 1: RMSE comparison results.

Position number	Markov	LSTM	D-Conv	STF-RNN	Social-LSTM	Social attention	DSCMP	BlindTPM
1	0.67	3.92	3.32	1.98	1.86	1.42	1.43	1.21
2	2.45	6.32	4.10	2.75	2.76	2.43	2.34	2.01
3	4.87	7.63	5.87	4.86	4.09	3.21	3.23	2.91
4	6.29	8.34	6.55	5.99	5.34	4.79	4.87	4.32
5	8.02	7.91	7.89	6.48	6.02	5.88	5.98	5.23

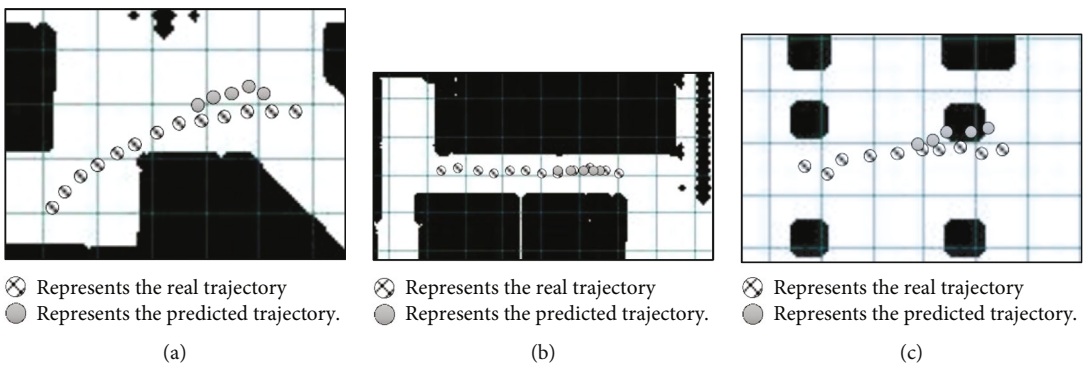


Figure 14: The results of BlindTPM prediction results in the coordinate system.

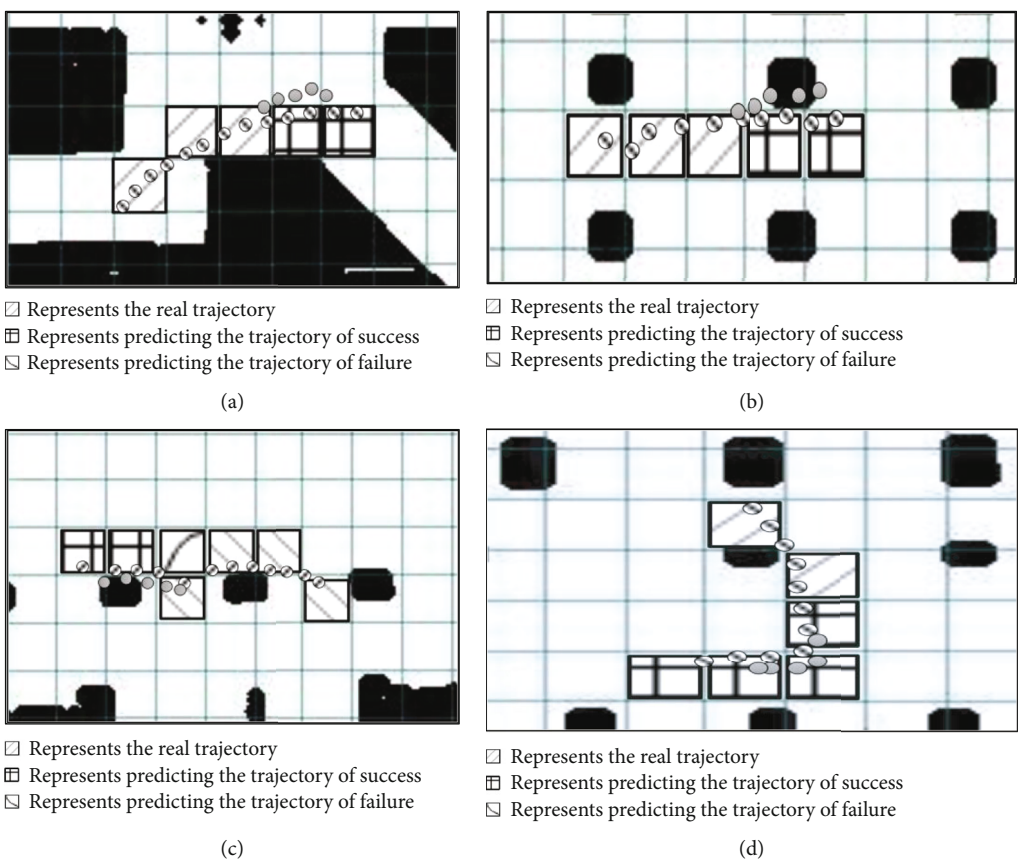


Figure 15: The results of BlindTPM prediction results in the obstacle distribution grid.

realizes the feature capture of the trajectory from the bottom to the top. The constantly changing size of the convolution kernel realizes the process of capturing spatial features with dynamic range changes. However, the nature of D-Conv is a convolution process that makes it cannot perform step-by-step timing-level derivation like recurrent networks. The

Table 2: The comparison results of prediction error.

Indicators	Markov	LSTM	D-Conv	STF-RNN	Social-LSTM	Social attention	DSCMP	BlindTPM (no estimation block)	BlindTPM
Accuracy	60%	72%	77%	82%	82%	83%	83%	84%	94%
Precision	59%	73%	80%	83%	79%	81%	84%	83%	93%
Recall	61%	71%	79%	81%	80%	80%	85%	84%	92%

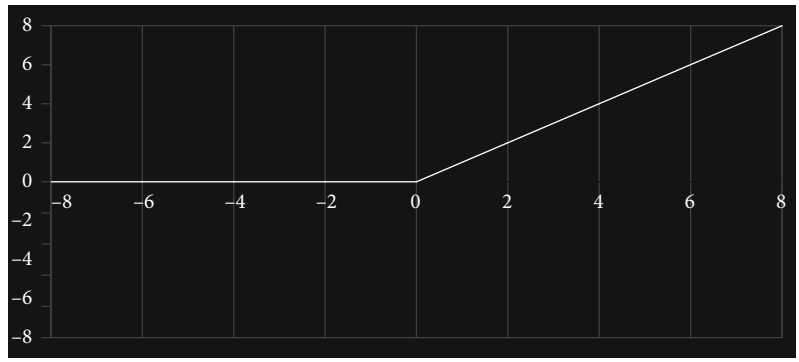


Figure 16: Activation function curve. The abscissa is the input value, and the ordinate corresponds to the output value of the input value. The output value of the negative interval is equal to 0, and the output value of the positive interval is the original value.

BlindTPM and other spatial-temporal models consider both the spatial correlation and temporal dependence of the trajectory. Therefore, they have shown a clear advantage in the forecast indicators of the next five points. The BlindTPM uses the same structure as D-Conv in the spatial correlation feature, which can dynamically obtain the spatial distribution weight of the historical trajectory. In the calculation process of temporal dependence, the BlindTPM can use all the historical information of the previous hidden layer to derive the parameter hidden state of the next layer.

Although the above model shows certain data advantages in RMSE, the predicted trajectory has the defects of dense location points and unavoidable obstacles in the actual scene, as shown in Figure 14. Figure 14(a) is an ideal prediction result, and its predicted value has the movement trend of the original trajectory. More importantly, its predicted value is scattered and does not appear in the obstacle area. (b) shows the prediction of a model of a straight trajectory, and it has the defect of too dense predictions. The reason for this phenomenon is that the spatial sample of the linear trajectory is single, which makes the weight of model parameters small. The predicted trajectory in (c) overlaps with the obstacle, because the modeling process does not consider the spatial distribution of the obstacle in the indoor scene. The experimental results of Figure 14 show that there are loopholes in the actual application effect of blind trajectory prediction based on coordinates. We should design additional methods to achieve decentralization of trajectory points to improve the accuracy of prediction. At the same time, we should take the spatial distribution of obstacles in the actual scene as an important reference for the trajectory generation process.

4.2.4. Obstacle Distribution Grid. Figure 15 shows the prediction results of the blind trajectory after the abscissa and ordinate are transformed into grid labels. More importantly, the above results add the spatial distribution of obstacles. (a) shows that this design mode improves the prediction accuracy and its prediction results are consistent with the real trajectory. (b) and (c) show that the model can avoid obstacles after adding the obstacle distribution. The predicted trajectory points do not penetrate obstacles. (d) shows that the model overcomes the shortcomings of the concentration of prediction results when the data is input into the model in the form of grid labels. The grid at the last point of the blind position is successfully obtained, and the prediction range of the model is broadened. Table 2 is the statistical results of the prediction error of the experimental models. The accuracy indicator represents the percentage of the correct results in the total sample. Although the accuracy indicator can judge the total accuracy rate, it cannot be used as a good indicator to measure the result when the sample is unbalanced. The precision indicator represents the probability of the samples that are actually positive among all the samples that are predicted to be positive. The precision indicator refers to the accuracy of the model's prediction of the results of positive samples, but the accuracy indicator refers to the overall prediction accuracy, which includes both positive samples and negative samples. The recall indicator represents the probability that a positive sample is predicted to be a positive sample. The higher the recall, the higher the probability that the actual negative sample will be predicted. Three indicators are aimed at the average value of the predicted 5 locations. We emphasize that the prediction and evaluation standard of accuracy is that only the grid labels of the abscissa and the ordinate can predict success at the same time. The BlindTPM has an excellent performance in three indicators and has a great improvement in accuracy (11%), precision (9%), and recall (7%) compared to the best performing model (DSCMP).

4.3. Analysis and Expansion. Mapping the coordinate system to grid labels allows us to get rid of the problem of insufficient

predictive ability of the neural network model for floating-point numbers, which also benefits from the invariance of indoor scenes. The grid label makes the trajectory prediction task transformed from a regression problem to a classification problem. The future location is composed of labels with a certain range, and the training process of the model has a clear goal to improve the accuracy of prediction. In the obstacle distribution grid, the area where the obstacle is located is assigned the value 0 and the passable area is assigned the value 1. The GCN weights the obstacle distribution grid and the trajectory distribution grid in the same dimension. After the activation function is calculated, the characteristics of the passable area are enhanced, and the zero-value characteristics of the obstacle area are discarded by the activation function. The activation function (ReLU) makes the feature of the passable area strengthen and the zero-value feature of the obstacle area is discarded, as shown in Figure 16. In order to verify the effect of our proposed estimation block, we set up an ablation experiment, as shown in Table 2. The prediction accuracy of BlindTPM has been significantly improved after adding the estimation block. This is because the local estimation increases the recursive correlation among the trajectory points of the blind. The temporary hidden state is generated by time series data through local estimation. These hidden states that can represent the blind person's short-term future location points are globally estimated to form high-level features that can represent the complete trajectory of the blind. Finally, the estimation block completes the task of improving the prediction accuracy.

5. Conclusions

We propose an obstacle avoidance algorithm and a trajectory prediction spatial-temporal model for the blind aiming at the auxiliary motion tasks of the blind in indoor scenes. Indoor scenes are transformed into grid data in this work. The trajectory of the blind person is accordingly transformed into grid labels. This conversion method limits the range of predicted values and overcomes the difficulty of predicting floating-point trajectory data by neural network models. We convert the obstacles in the scene into the obstacle spatial distribution matrix according to certain rules to reduce the weight of the area where the obstacle is located and increase the weight of the passable area. Experiments verify the difference between coordinate trajectory and grid trajectory, which proves the superiority of the design mode of this work. In future work, we expect that the influence of other populations on the trajectory of blind people will be added to existing research. Its necessity is reflected in the fact that humans are thinking creatures and their actions are always affected by the surrounding environment [29] or other humans to act accordingly.

Data Availability

The dataset of this work comes from the research of Kacorri et al. [26] which is about the movement of blind people indoors. The URL of the original dataset is at https://envfactors.github.io/. We perform desensitization processing and obstacle grid processing on the original dataset.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported in part by the National Key Research and Development Plan Key Special Projects under Grant No. 2018YFB2100303, Shandong Province Colleges and Universities Youth Innovation Technology Plan Innovation Team Project under Grant No. 2020KJN011, Shandong Provincial Natural Science Foundation under Grant No. ZR2020MF060, Program for Innovative Postdoctoral Talents in Shandong Province under Grant No. 40618030001, National Natural Science Foundation of China under Grant No. 61802216, and Postdoctoral Science Foundation of China under Grant No. 2018M642613.

References

[1] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2306–2320, 2017.

[2] L. Li, B. Zhou, J. Lian, and Y. Zhou, "Research on pedestrian trajectory prediction method based on social attention mechanism," *Journal on Communications*, vol. 41, no. 6, pp. 175–183, 2020.

[3] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[4] P. Trautman and A. Krause, "Unfreezing the robot: navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, Taipei, Taiwan, October 2010.

[5] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2287–2301, 2011.

[6] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7575*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., pp. 201–214, Springer, Berlin, Heidelberg, 2012.

[7] P. Dayan and B. W. Balleine, "Reward, motivation, and reinforcement learning," *Neuron*, vol. 36, no. 2, pp. 285–298, 2002.

[8] Z. Y. Zhang and Y. H. Diao, "Pedestrian trajectory prediction model with social features and attention," *Journal of Xidian University (Natural Science)*, vol. 47, no. 1, pp. 10–17, 2020.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] K. Xu, Z. Qin, G. Wang, K. Huang, S. Ye, and H. Zhang, "Collision-free lstm for human trajectory prediction," in *MultiMedia Modeling. MMM 2018. Lecture Notes in Computer Science, vol 10704*, pp. 106–116, Springer, Cham, 2018.

[11] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9907*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 816–833, Springer, Cham, 2016.

[12] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, Las Vegas, NV, USA, June 2016.

[13] A. Vemula, K. Muelling, and J. Oh, "Social attention: modeling attention in human crowds," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4601–4607, Brisbane, QLD, Australia, May 2018.

[14] V. Todi, G. Sengupta, and S. Bhattacharya, "Probabilistic path planning using obstacle trajectory prediction," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data.*, pp. 36–43, Kolkata India, 2019.

[15] Z. Wang, M. L. Fang, and X. N. Lin, "Route planning in handheld blind navigation system based on RFID," *Computer Engineering and Design*, vol. 33, no. 5, pp. 2063–2067, 2012.

[16] M. Luo, X. Hou, and J. Yang, "Surface optimal path planning using an extended Dijkstra algorithm," *IEEE Access*, vol. 8, no. 1, pp. 147827–147838, 2020.

[17] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6861–6871, 2019.

[18] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, http://arxiv.org/abs/1511.08458.

[19] L. Zhao, Y. Song, C. Zhang et al., "T-gcn: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[20] W. Sun and X. Zhou, "Characterization of local sampling sequences for spline subspaces," *Advances in Computational Mathematics*, vol. 30, no. 2, pp. 153–175, 2009.

[21] Z. Lv, J. Li, C. Dong, and W. Zhao, "A deep spatial-temporal network for vehicle trajectory prediction," in *Wireless Algorithms, Systems, and Applications. WASA 2020. Lecture Notes in Computer Science, vol 12384*, D. Yu, F. Dressler, and J. Yu, Eds., pp. 359–369, Springer, Cham, 2020.

[22] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016, http://arxiv.org/abs/1606.09375.

[24] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, http://arxiv.org/abs/1803.01271.

[25] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? Estimating travel time based on deep neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[26] H. Kacorri, E. Ohn-Bar, K. M. Kitani, and C. Asakawa, "Environmental factors in indoor navigation based on real-world trajectories of blind users," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, Montreal QC Canada, 2018.

[27] A. Al-Molegi, M. Jabreel, and B. Ghaleb, "STF-RNN: space time features-based recurrent neural network for predicting people next location," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, Athens, Greece, December 2016.

[28] C. Tao, Q. Jiang, L. Duan, and P. Luo, "Dynamic and static context-aware LSTM for multi-agent motion prediction," pp. 547–563, 2020, http://arxiv.org/abs/2008.00777.

[29] K. Mangalam, H. Girase, S. Agarwal et al., "It is not the journey but the destination: endpoint conditioned trajectory prediction," in *Proceedings of the 16th European Conference on Computer Vision*, pp. 759–776, Glasgow, UK, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9249387, 11 pages
https://doi.org/10.1155/2021/9249387

Research Article

Deep Learning Enhanced Solar Energy Forecasting with AI-Driven IoT

Hangxia Zhou,[1] **Qian Liu**,[1] **Ke Yan**,[2] **and Yang Du**[3]

[1]*Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou, China 310018*
[2]*National University of Singapore, 4 Architecture Drive, Singapore 117566*
[3]*College of Science and Engineering, James Cook University, Cairns, QLD 4870, Australia*

Correspondence should be addressed to Qian Liu; liuqian235@cjlu.edu.cn

Received 14 April 2021; Accepted 7 June 2021; Published 19 June 2021

Academic Editor: Xiao Zhang

Short-term photovoltaic (PV) energy generation forecasting models are important, stabilizing the power integration between the PV and the smart grid for artificial intelligence- (AI-) driven internet of things (IoT) modeling of smart cities. With the recent development of AI and IoT technologies, it is possible for deep learning techniques to achieve more accurate energy generation forecasting results for the PV systems. Difficulties exist for the traditional PV energy generation forecasting method considering external feature variables, such as the seasonality. In this study, we propose a hybrid deep learning method that combines the clustering techniques, convolutional neural network (CNN), long short-term memory (LSTM), and attention mechanism with the wireless sensor network to overcome the existing difficulties of the PV energy generation forecasting problem. The overall proposed method is divided into three stages, namely, clustering, training, and forecasting. In the clustering stage, correlation analysis and self-organizing mapping are employed to select the highest relevant factors in historical data. In the training stage, a convolutional neural network, long short-term memory neural network, and attention mechanism are combined to construct a hybrid deep learning model to perform the forecasting task. In the testing stage, the most appropriate training model is selected based on the month of the testing data. The experimental results showed significantly higher prediction accuracy rates for all time intervals compared to existing methods, including traditional artificial neural networks, long short-term memory neural networks, and an algorithm combining long short-term memory neural network and attention mechanism.

1. Introduction

Photovoltaic power generation has the advantages of low carbon consumption, adaptive to various applications, and low installation and maintenance costs, which is known as a sustainable energy source [1]. Because of different weather conditions, PV panels often cannot stably output electrical power from solar energy. While integrating the PV power to the power grid, the grid is seriously influenced. The stability of the entire grid will be greatly reduced. With the wireless sensor network, it is possible for deep learning techniques to forecast solar energy generation and consequently stabilize the smart grid systems [2]. Therefore, artificial intelligence- (AI-) driven internet of things (IoT) technology becomes one of the key technologies of solving this problem [3, 4]. Along with the fast development of IoT technology, the extended deep learning methods nowadays are capable of performing short-term time series data forecasting, such as PV power generation, anomaly detection, and energy consumption forecasting, with considerable high forecasting accuracy [5–7].

In recent years, with the fast development of AI-driven IoT technology, the applications of deep learning technologies have been extended to various fields [8, 9], such as digital twinning [10], computer security [11], cyberphysical systems [12], transportation systems [13], and air quality forecasting [14]. Jiménez-Pérez and Mora-López [15] proposed a forecasting system simulating global solar irradiance forecasts every hour. The system can be separated into two phases. The first phase is clustering. The original dataset was divided

into groups by the k-means clustering algorithm. Each group represents data of different weather types. Machine learning techniques, such as decision trees, artificial neural networks, and support vector machines, are employed to perform forecasting in the second phase. Yang et al. [16] proposed a hybrid deep learning method based on weather types for PV power output forecasting with a timestep at 1 hour. The proposed method has three steps: classification, training, and prediction. The classification step involves a self-organizing map (SOM) [17] and Learning Vector Quantization (LVQ) [18]. In the training and prediction step, the fuzzy training method is used to select the most appropriate deep learning model for prediction. Han et al. [19] proposed an alternative multimodel PV power interval prediction method that considers the seasonal property of the PV power and absolute power deviation in the prediction process. Van-Deventer et al. [20] proposed a support vector machine model based on a genetic algorithm (GASVM). The GASVM model classifies the historical weather data using an SVM classifier initially, and later, it is optimized by the genetic algorithm using an ensemble technique. Malvoni et al. [21] proposed to use wavelet decomposition and principal component analysis to decompose meteorological data and treat it as predictive inputs. A time series forecasting method, named as GLSSVM (Group Least Square Support Vector Machine), which combines the Least Square Support Vector Machines (LS-SVM) and Group Method of Data Handling (GMDH), was applied to the measured weather data.

More recently, deep learning techniques, such as the convolutional neural network (CNN), long short-term memory (LSTM), and deep belief networks (DBNs), are widely applied to the PV system energy generation forecasting problem. Srivastava and Lessmann [22] proposed to adopt the original LSTM neural network for PV energy generation forecasting, and the proposed algorithm is hardly generalizable. Jian et al. [23] proposed an online sequential extreme learning machine with a forgetting mechanism model for PV energy generation prediction. The normalized data is input into a two-layer CNN for feature extraction and then merged into LSTM by fusion layer for PV power prediction. Zhou et al. [24] proposed a LSTM neural network combining the attention mechanism (AM) for PV energy generation forecasting. The AM is used to adaptively perceive feature information from the time series data. Chen et al. [25] proposed a cloud shadow model using computer vision techniques to forecast real cloud coverage nature for PV systems and enhanced the PV energy generation forecasting results consequently. Chang et al. [26] introduced a virtual inertia control based on PV load forecasting results, showing real-world applications of the PV energy generation forecasting techniques.

In this paper, we proposed a novel PV power forecasting framework combining clustering and deep learning technologies. The entire framework can be divided into three stages. In the first stage, according to the impact of the training data in the previous paragraph, we divide the training data into four clusters using the SOM algorithm, mimicking the four seasons. The reason for using the SOM algorithm as the experimental clustering algorithm is described in 2.2.2, and the reason for simulating the data into four seasons for the experiment is described in 2.2.1. In the second stage, use AM, CNN, and LSTM to build the model. The third stage is the forecasting stage. According to the month of the testing data, the most suitable forecasting model is selected to predict the PV energy generation for the next time stamp. Experimental results show the superior performance of the proposed method over the existing works. Compared to the existing methods tackling on the same problem, the main contributions of the proposed method are summarized in the following points.

1.1. The Raw Data Is Processed by Clustering Techniques for More Accurate Forecasting Performance. Based on the data collected from the eastern region of China, the solar irradiance patterns vary between different seasons. The data is first clustered into four classes and trained using four different LSTM neural networks, respectively, to enhance the forecasting accuracy.

1.2. CNN and LSTM Are Combined for a More Sophisticated Deep Learning Framework. Both CNN and LSTM are popular deep learning techniques for forecasting problems. In this study, the two techniques are combined to produce better forecasting results for the PV energy generation forecasting problem.

1.3. The AM Technique Is Adopted to Further Enhance the Forecasting Performance. The AM is an important extension for the traditional LSTM neural network. It forces the LSTM neural networks to pay more attention to the features that are more relevant to the forecasting outputs. According to the experimental results, the AM technique greatly improves the forecasting results produced by LSTM neural networks.

2. Methodology

The dataset employed in this study was collected by a PV power station located in Shaoxing city in the eastern part of China. This solar energy generation dataset was collected from October 2014 to September 2018, in a time interval of 7.5 min. The data from 2014 to 2016 is taken as the training dataset, and the data from 2017 to 2018 is used for testing. Remote sensors were utilized to record the PV module temperature, the current, voltage, frequency, phases, and PV power every 7.5 minutes. Since the power station uses three-phase inverter equipment, the data contains the PV module temperature, three AC currents (alternating current 1, alternating current 2, and alternating current 3), three AC voltages (AC voltage 1, AC voltage 2, and AC voltage 3), two DC currents (direct current 1 and direct current 2), two DC voltages (DC voltage 1 and DC voltage 1), frequency, phases, and PV power. The frequency and phase do not change with the PV power, so the frequency and phase data are not used in the experiments. It is very difficult to obtain huge amount of historical weather data with a time interval of 7.5 min at the same location from the weather bureau. We do not add weather data to the experimental data and find a data preprocessing method to effectively improve the predictive ability of the model.

We calculated correlation coefficients between other factor data and PV power. The correlation of the multifactor combination on the PV power is shown in Table 1. A combined correlation test method for calculating the modified RV coefficient is introduced here. Modified RV coefficient (RV_{mod}) is a correlation analysis method based on matrix calculation. The equation for calculating the modified RV coefficient is

$$RV_{mod} = \frac{tr(JK)}{\sqrt{tr(J)^2 tr(K)^2}}, \quad (1)$$

where $J = M \cdot M\,T - \text{diag}\ (M \cdot M\,T)$; $K = N \cdot N\,T - \text{diag}\ (N \cdot N\,T)$; M represents the matrix of influencing factors, including alternating current 2 (AC2) and alternating current 3 (AC 3); N represents PV power output matrix; diag $(\cdot)$ is a function that takes out the diagonal elements of the matrix; $tr(\cdot)$ is a function that takes the sum of the elements on the diagonal of the matrix; the range of RV_{mod} is (-1,1), when the RV_{mod} is closer to -1 or 1; and the correlation is higher between the factor and the power output.

For the coefficients AC2 and coefficients AC3 which are very close, we specially carry out experiments and compare the experimental results of the datasets of AC2 and AC3. The experimental results show that, although the coefficient difference is not large, the prediction accuracy of AC2 is significantly higher than AC3. In this study, we extract AC2 data as experiment data.

2.1. Clustering of the Original Data. In the clustering stage, the AC2 data from 2014 to 2016 is the training dataset and used for clustering. Considering the different numbers of days per month and the lack of PV data, we selected 15 days of complete data from each month and made each month AC2 data to a dataset as $[x_1, x_2, \cdots, x_{15}]$; the dataset indicates that a month's AC2 data, x, contains 192 AC2 real-time data for one day. The clustering experiment uses two-year AC2 data. We merged the datasets with the same month label. The SOM algorithm is used to perform the clustering task. The raw data was clustered into 4 clusters.

The SOM model structure is shown in Figure 1. The learning steps of the SOM algorithm are as follows.

In the first step, the weight vector corresponding to each neuron in the competition layer was initialized, normalizing the current input mode vector X and the weight vector corresponding to the neuron. Secondly, the neuron node d with the smallest Euclidean distance between X_c (input vector) and $\boxtimes_j$ (connection weight vector) is selected as the winning neuron node. As a result, the weights were updated accordingly. Following Equation (3), the weight of the winning neuron node d and other nodes in its domain is updated, where $\eta(t)$ is the learning rate of the t step in the range of (0, 1) and $h_{d\boxtimes j}(t)$ is the neighborhood function. $h_{d\boxtimes j}(t)$ generally uses a Gaussian function, as shown in Equation (4). The specific adjustment rules are as shown in Equations (5) and (6).

$$\left\|X_c - w_j\right\| = \min\left\{d_{jc}\right\}, \quad (2)$$

$$W_{ij}(t+1) = W_{ij}(t) + \eta(t) h_{d,j}(t)\left(X_i - W_{ij}(t)\right), \quad (3)$$

$$h_{d,j}(t) = \exp\left(-\frac{d_{dj}^2}{2r^2(t)}\right), \quad (4)$$

$$r(t+1) = \text{INT}\left((r(t)-1) \times \left(1 - \frac{t}{T}\right)\right) + 1, \quad (5)$$

$$\eta(t+1) = \eta(t) - \frac{\eta(0)}{T}, \quad (6)$$

where the distance between the neuron d and the neuron j is ${d_{dj}}^2$, $r(t)$ is the neighborhood radius, INT is the rounding function, and T is the learning frequency.

2.2. Deep Learning Models. In the training stage, the 4 clusters of data are inputted into a model for training, and 4 forecasting models are trained. The training process follows the training method of time series. We display the training data of each category as $[y_1, y_2, y_3, \cdots, y_n]$, where y represents AC2 data and n represents the number of AC2 data contained in each category. We set timestep i, $[y_1, y_2, \cdots, y_i]$, as a new training set, and $y_i + 1$ as the target value; then, we put the new training set into the training model, and the model parameters are adjusted during training so that the predicted output is constantly close to $y_i + 1$. Next, $[y_2, y_3, \cdots, y_i + 1]$ is inputted into the training model and adjusts the model parameters so that the predicted output is continuously close to $y_i + 2$, and so on, until the target value is y_n; end the model training and get the forecasting model.

The training model structure of this experiment consists of two convolutional kernel layers of CNN, one layer of LSTM with attention mechanism, and a fully connected (FC) layer. The overall flowchart of the proposed framework is depicted in Figure 2.

This proposed deep learning model (CNN-ALSTM) is a hybrid deep learning model to extract features from the raw data and perform foresting using the LSTM neural network. The CNN layer is employed to extract the useful features from the time series data, representing additional latent information of the data, which has the potential to improve the prediction accuracy. The experimental results show that the CNN layer contains one 16 3×1 convolution kernel layer and one 32 3×1 convolution kernel convolution layer that optimizes the prediction performance.

The feature vector obtained from the second layer of CNN was inputted into the LSTM layer for prediction. Each element of the feature vector has correspondence to one of the 32 units in the LSTM layer. The attention mechanism puts higher weight to the feature quantities that are significantly related to the current output. Last, the output vector of the attention mechanism is processed by a FC layer using the unfolding operation. The predicted value of the AC2 at the next moment is output.

LSTM is very suitable for prediction experiments of time series data. Existing works show superior forecasting performance combining CNN and LSTM for various applications

Table 1: Correlation coefficient of PV data factors.

PV data factor name	Correlation coefficient
DC voltage 1 (V)	0.19
DC voltage 2 (V)	0.20
Direct current 1 (A)	0.13
Direct current 2 (A)	0.13
AC voltage 1 (V)	0.10
AC voltage 2 (V)	0.11
AC voltage 3 (V)	0.14
Alternating current 1 (A)	0.28
Alternating current 2 (A)	0.992
Alternating current 3 (A)	0.998
Alternating current 2 (A) and alternating current 3 (A) combination	0.73
PV module temperature	0.16

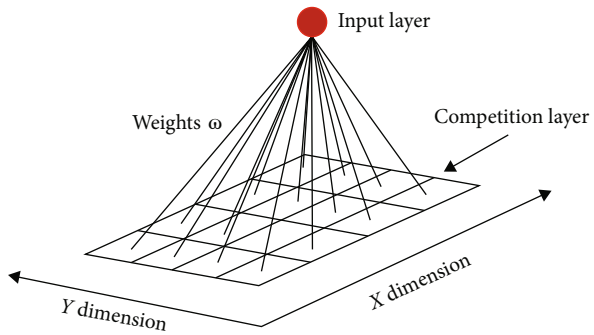


Figure 1: Structure of the SOM model.

[23, 27–29]. CNN helps LSTM to better extract the characteristics of experimental data. The attention mechanism is a process of assigning weights. Using the attention mechanism, more accurate weightage values are assigned to the LSTM output vector to improve the prediction capability of the model.

The predictive power is obtained by the inverse normalization according to Equation (7).

$$\mathrm{Pr}_P = \mathrm{Pr}_{I\mathrm{ac2}} * (P_{\max} - P_{\min}) + P_{\min}, \tag{7}$$

where Pr_p is the predicted value of power and $\mathrm{Pr}_{I\mathrm{ac2}}$ is the predicted value of AC2.

2.2.1. Long Short-Term Memory Neural Network. The LSTM model structure is shown in Figure 3.

The appearance of LSTM cell structure effectively resolves the gradient explosion/vanishing problems. There are four important elements in the flowchart of the LSTM model: cell status, input gate, forget gate, and output gate. The input, forget, and output gates are used to control the update, maintenance, and deletion of information contained in cell status. The forward computation process can be denoted as

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ O_t &= \sigma(W_O \cdot [h_{t-1}, x_t] + b_o), \\ \tilde{C}_t &= \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right), \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\ h_t &= O_t \cdot \tanh\left(C_t\right), \end{aligned} \tag{8}$$

where W_f, W_i, and W_o are the weight matrix of the forgetting gate, input gate, and output gate, respectively; b_f, b_i, and b_o are the offset items of the forgetting gate, the input gate, and the output gate, respectively; σ is the sigmoid activation function; tanh is the hyperbolic tangent activation function.

2.2.2. The Attention Mechanism. The attention mechanism is a brain signal processing mechanism peculiar to human vision. Human vision quickly scans the global image to obtain the target area that needs attention and ignores other areas of useless information. The attention mechanism algorithm has been successfully implemented and applied to model training [26] and other related fields.

The model proposed in this paper uses the LSTM hidden layer output vector $H = \{h_1, h_2, \cdots, h_t\}$ as the input of the attention mechanism, and the attention mechanism will find the attention weight α_i of h_i, which can be calculated as shown in

$$\begin{aligned} e_i &= \tanh\left(W_h h_i + b_h\right), \\ \alpha_i &= \frac{\exp\left(e_i\right)}{\sum_{i=1}^{t} \exp(e_i)}, \end{aligned} \tag{9}$$

where W_h is the weight matrix of h_i and b_h is the bias.

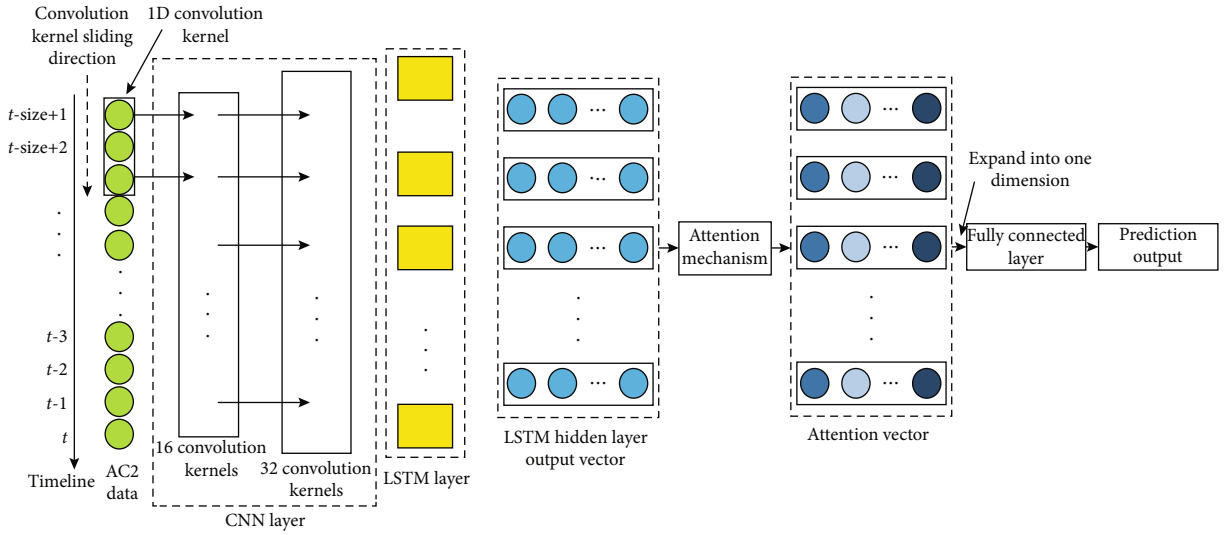


Figure 2: The overall flowchart of the proposed algorithm.

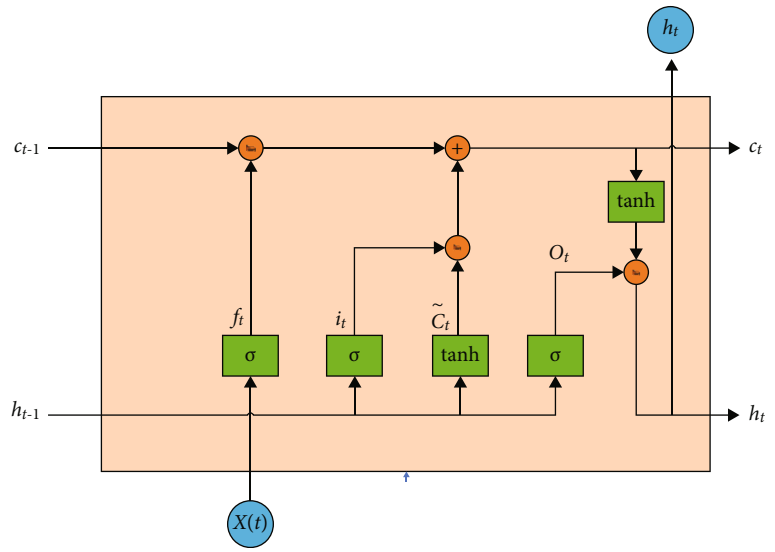


Figure 3: Structure of LSTM.

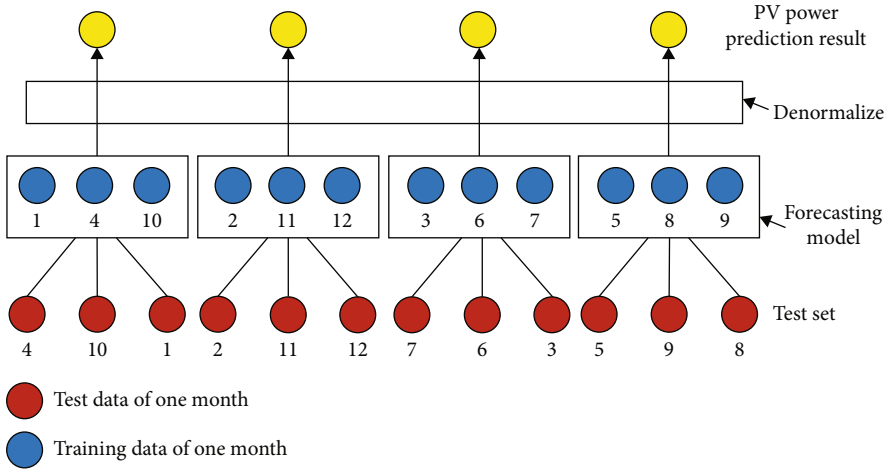


Figure 4: Structure of prediction stage. The numbers under test data and training data represent months.

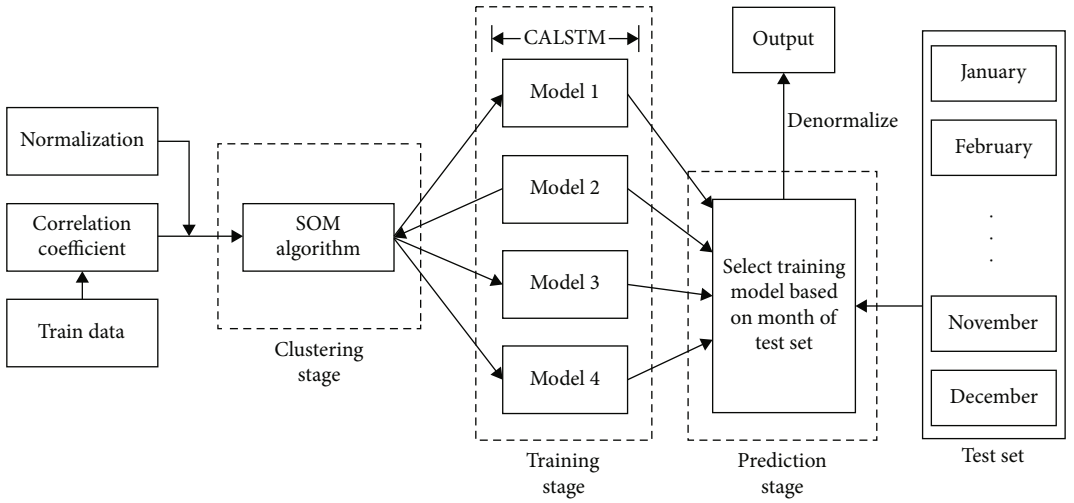


Figure 5: Structure of the proposed hybrid model.

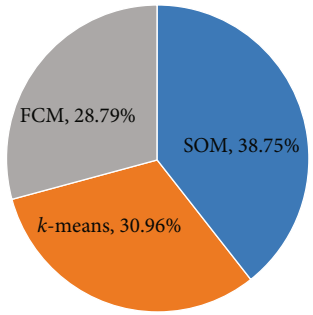


Figure 6: Accuracy of different clustering algorithms in the training phase.

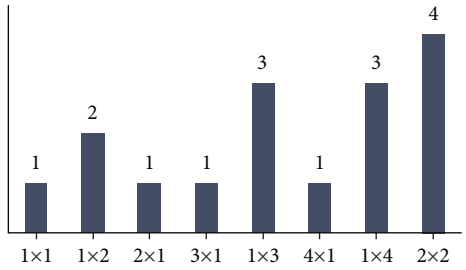


Figure 7: Horizontally, the X-axis represents the size of the competition layer, and vertically, the Y-axis represents the number of output neurons.

The values of W_h and b_h vary during the ALSTM training process. The range of e_i is (-1,1). The attention vector $H' = \{h_1', h_2', \cdots, h_t'\}$ can be obtained by multiplying attention weight α_i and h_i:

$$h_i' = \alpha_i \cdot h_i. \tag{10}$$

The attention mechanism is implemented as a custom layer where the parameters are optimized using RMSProp backpropagation [33, 34].

2.3. The Overall Forecasting Framework. There are four deep learning models trained in the proposed framework (Figure 4). In the testing stage, an appropriate forecasting model is selected based on the grouping result of the testing data sample. The proposed hybrid deep learning model structure is shown in Figure 5. As a necessary preprocessing step, we used correlation coefficient to determine inputs for prediction. The weather data was employed by the SOM algorithm to cluster inputs into four categories. Each category is trained by the proposed hybrid deep learning framework. In particular, the CNN is used to extract the time series features of AC2 data; the LSTM neural network further extracts high latitude features in the data. The attention mechanism is used to assign different attention weights to the output elements of the LSTM hidden layer. In the testing phase, according to the month of the test set, we chose the appropriate model to predict the testing result (Figures 4 and 5).

The process of data classification method determination and hyperparameter determination is also described, including the reason for simulating data in four seasons, the selection of clustering algorithm, the settings of SOM competition layer, and the selection of SOM classified result.

2.3.1. The Selection of Clustering Algorithm. The original idea of the clustering is to map the raw data into four different groups, representing the four seasons. These four groups of data are trained and tested using the proposed CNN-ALSTM structure separately. Different clustering algorithms are tested in the section. In the training phase, we analyze clustering algorithms, including k-means, FCM, and SOM (Figure 6), for clustering the training dataset into four clusters. Based on the experimental results on the training data, SOM is more accurate than most of the above-mentioned traditional clustering algorithms (Figure 6) according to the original season labels of the raw data.

The size of the SOM competition layer has been evaluated through a series of experiments as shown in Figure 7. The X-axis represents the size of the competition layer, and

Table 2: Predicting MAPE results for one year of data.

		2017.10	2017.11	2017.12	2018.1	2018.2	2018.3	2018.4	2018.5	2018.6	2018.7	2018.8	2018.9	Overall
7.5 min	MLP	22.15	20.07	23.05	25.62	18.10	24.97	26.00	23.19	28.39	25.08	22.18	20.03	23.11
	LSTM	23.87	21.42	20.96	28.76	16.63	28.96	20.99	31.19	40.02	20.21	22.18	21.73	24.61
	ALSTM	21.64	22.44	21.44	28.16	23.52	24.55	18.43	17.82	32.79	25.46	29.05	19.78	23.32
	Prop.	19.95	16.76	20.79	25.01	16.90	19.29	15.84	16.83	17.21	22.20	19.88	18.69	18.82
15 min	MLP	30.19	24.41	26.42	28.58	20.63	37.38	47.40	40.80	53.70	38.39	31.30	27.10	34.17
	LSTM	23.42	21.47	23.74	27.51	17.74	23.62	22.63	21.21	28.95	28.72	19.00	18.64	22.75
	ALSTM	20.30	23.08	22.40	28.38	20.40	29.96	29.69	20.53	68.64	47.60	24.06	22.49	28.82
	Prop.	22.69	20.32	26.51	28.04	21.055	28.17	28.61	25.12	18.23	25.44	23.87	20.89	23.79
30 min	MLP	33.09	35.15	36.98	37.68	28.49	33.44	28.85	28.92	30.17	35.92	32.78	30.65	32.57
	LSTM	37.75	41.62	38.86	38.42	41.45	36.07	36.34	30.06	37.90	39.01	32.59	40.96	37.39
	ALSTM	25.67	29.86	37.76	42.77	33.89	30.41	57.01	28.64	30.09	30.71	37.50	35.79	34.18
	Prop.	32.62	29.59	29.32	32.88	23.50	33.38	34.87	31.34	45.62	30.19	31.84	32.62	32.39

Table 3: Predicting RMSE results for one year of data.

		2017.10	2017.11	2017.12	2018.1	2018.2	2018.3	2018.4	2018.5	2018.6	2018.7	2018.8	2018.9	Overall
7.5 min	MLP	1.39	1.58	2.09	2.16	1.23	1.07	0.63	0.72	0.69	1.25	1.10	1.45	1.32
	LSTM	1.45	1.63	2.16	2.20	1.29	1.10	0.63	0.81	0.78	1.30	1.15	1.50	1.37
	ALSTM	1.60	1.75	2.16	2.36	1.33	1.10	0.69	0.81	0.71	1.34	1.17	1.63	1.42
	Prop.	1.44	1.52	2.08	2.32	1.23	1.10	0.63	0.65	0.70	1.36	1.24	1.40	1.30
15 min	MLP	1.42	1.60	2.12	2.19	1.22	1.12	0.77	0.82	0.82	1.29	1.16	1.47	1.37
	LSTM	1.63	1.94	2.51	2.44	1.45	1.16	0.69	0.83	0.73	1.56	1.17	1.61	1.53
	ALSTM	1.54	1.90	2.38	2.49	1.74	1.19	0.71	0.77	0.96	1.61	1.25	1.69	1.55
	Prop.	1.57	1.78	1.47	1.86	1.47	1.21	0.75	0.90	0.87	1.45	1.29	1.62	1.40
30 min	MLP	2.23	2.74	2.91	3.29	1.84	1.56	0.94	1.22	1.10	1.96	1.83	2.03	2.02
	LSTM	3.10	3.95	4.13	3.72	3.52	2.01	1.46	1.64	1.19	2.32	2.52	3.51	2.89
	ALSTM	1.80	2.95	3.78	3.91	2.66	1.19	0.94	0.99	0.76	2.06	1.46	3.18	2.22
	Prop.	2.26	2.29	2.95	3.12	1.71	1.92	1.00	0.94	0.96	1.71	1.69	2.71	2.04

Table 4: Predicting MAE results for one year of data.

		2017.10	2017.11	2017.12	2018.1	2018.2	2018.3	2018.4	2018.5	2018.6	2018.7	2018.8	2018.9	Overall
7.5 min	MLP	0.829	0.88	1.15	1.36	0.70	0.61	0.39	0.42	0.39	0.72	0.67	0.83	0.72
	LSTM	0.89	0.95	1.21	1.47	0.74	0.67	0.35	0.52	0.53	0.74	0.69	0.91	0.78
	ALSTM	1.049	1.12	1.22	1.56	0.84	0.62	0.39	0.45	0.44	0.83	0.77	1.05	0.83
	Prop.	0.85	0.84	1.25	1.55	0.78	0.57	0.32	0.40	0.33	0.81	0.80	0.88	0.70
15 min	MLP	0.91	0.92	1.17	1.38	0.69	0.73	0.58	0.58	0.61	0.84	0.75	0.91	0.82
	LSTM	1.05	1.25	1.61	1.61	0.97	0.67	0.41	0.49	0.42	0.97	0.70	1.02	0.90
	ALSTM	0.95	1.24	1.42	1.65	1.25	0.76	0.47	0.44	0.78	1.16	0.82	1.12	0.95
	Prop.	1.022	1.20	1.53	1.55	1.16	0.74	0.49	0.54	0.42	0.81	0.83	0.91	0.85
30 min	MLP	1.58	2.02	2.09	2.29	1.35	1.03	0.55	0.73	0.57	1.26	1.30	1.47	1.32
	LSTM	2.18	2.98	3.00	2.73	2.78	1.27	0.86	0.97	0.67	1.51	1.73	2.60	1.90
	ALSTM	1.19	2.24	2.99	3.16	2.23	0.76	0.77	0.66	0.46	1.39	1.12	2.43	1.47
	Prop.	1.64	1.66	2.07	2.29	1.30	1.39	0.67	0.64	0.68	1.10	1.23	2.13	1.38

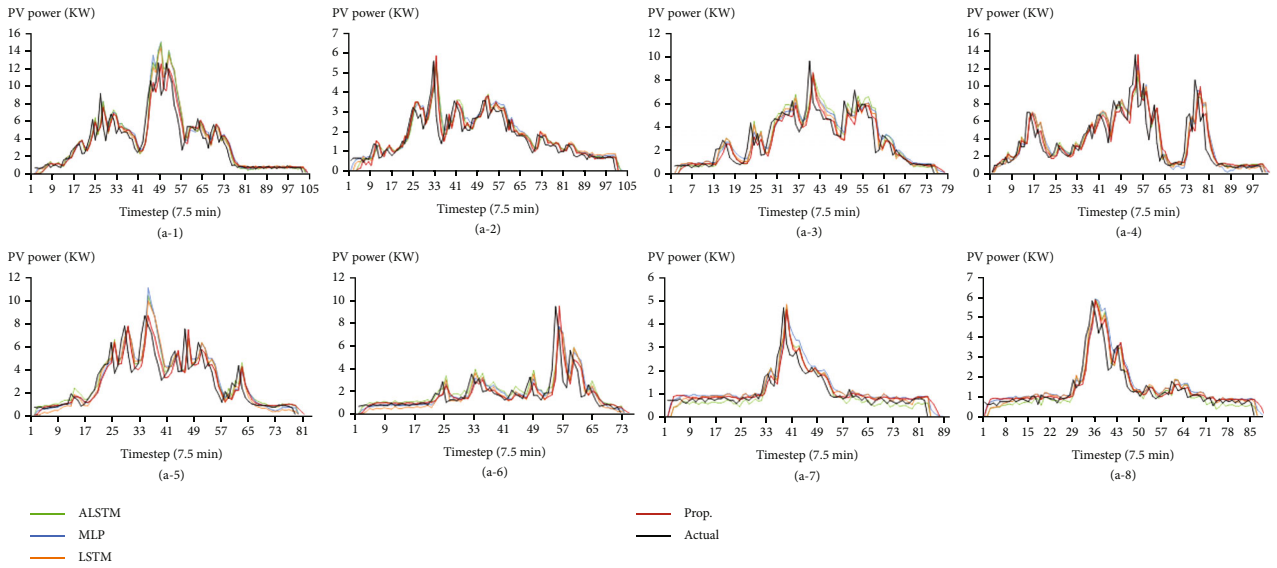


Figure 8: PV power forecast result in 7.5 min ahead. (a-1) and (a-2) are the January 5th and January 6th, (a-3) and (a-4) are the March 1st and March 3rd, (a-5) and (a-6) are the June 6th and June 8th, and (a-7) and (a-8) are the October 3rd and October 4th.

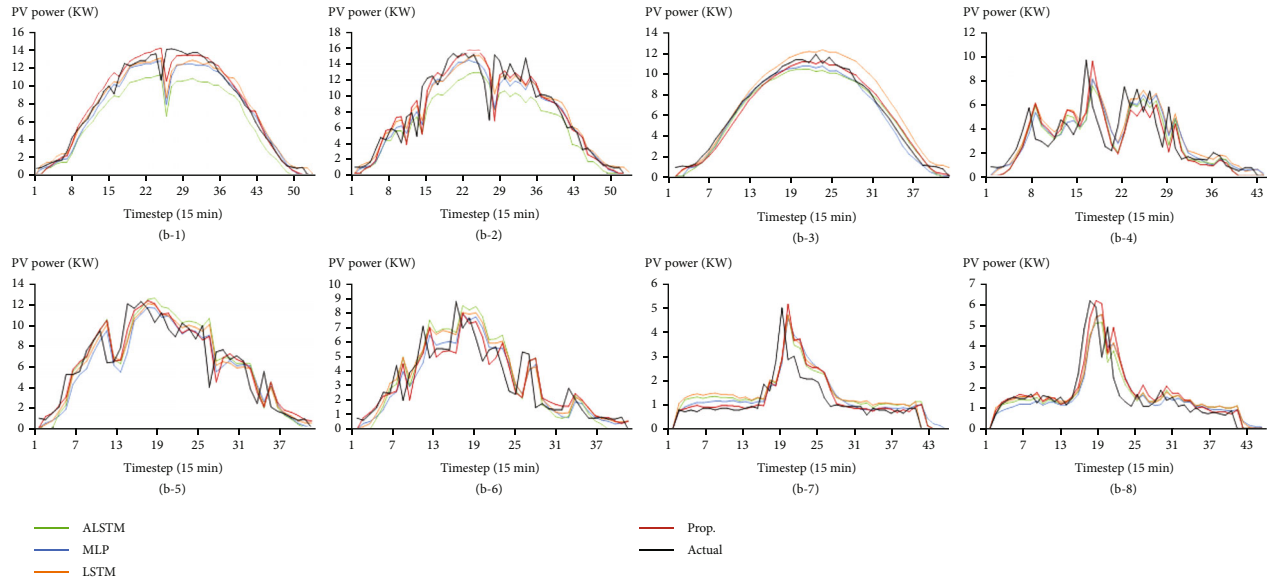


Figure 9: PV power forecast result in 15 min ahead. (b-1) and (b-2) are the January 14th and January 15th, (b-3) and (b-4) are the May 1st and May 2nd, (b-5) and (b-6) are the July 1st and July 2nd, and (b-7) and (b-8) are the October 3rd and October 4th.

the Y-axis represents the number of output neurons. The result shows that the most suitable competition layer for SOM is 2×2.

The clustering results of the SOM algorithm have a potential relationship with the prediction effect of the forecasting model. The number of iterations of the SOM algorithm is set between 500 and 1000, and the number of times is debugged in hundred. The batch size is set to 15 or 30.

3. Experimental Process and Results

In this study, the average absolute percentage error MAPE, the mean absolute error MAE, and the root mean square error RMSE are used to evaluate the prediction ability of the model. The detailed equations of three error metrics are formulated in

$$\begin{aligned} \text{MAPE} &= \frac{1}{n}\sum_{i=1}^{n}\frac{|x_{\text{model},i} - x_{\text{actual},i}|}{x_{\text{actual},i}}, \\ \text{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{\text{model},i} - x_{\text{actual},i})^2}, \\ \text{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|x_{\text{model},i} - x_{\text{actual},i}\right|. \end{aligned} \tag{11}$$

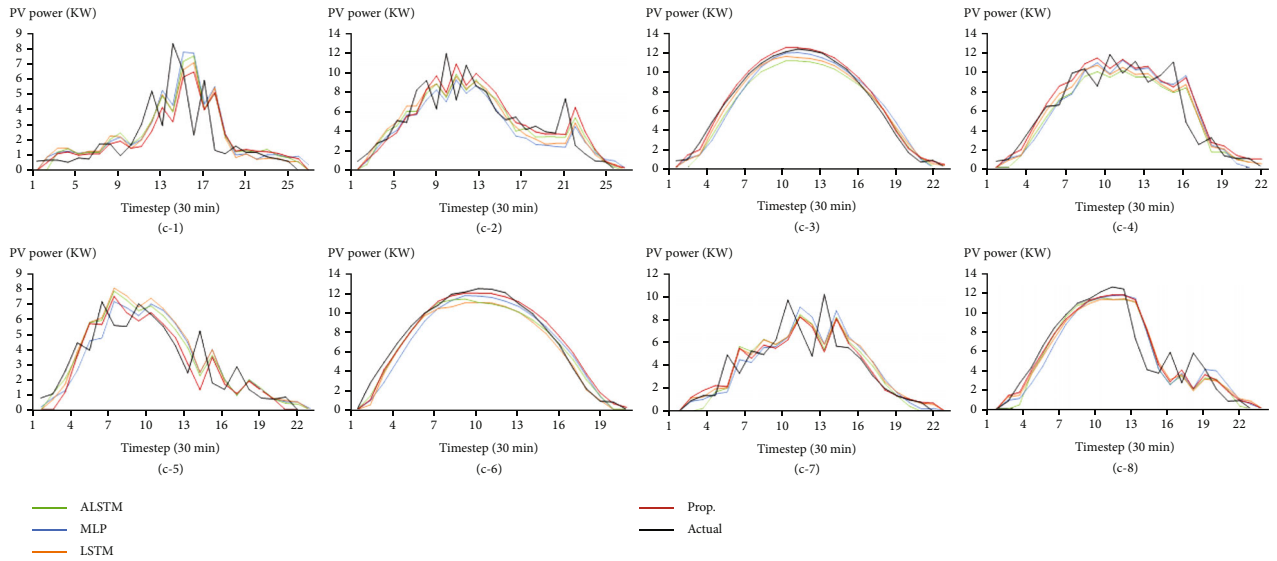


Figure 10: PV power forecast result in 30 min ahead. (c-1) and (c-2) are the January 9th and January 11th, (c-3) and (c-4) are the May 30th and May 31th, (c-5) and (c-6) are the August 2nd and July 9th, and (c-7) and (c-8) are the October 6th and October 12th.

The experiment compares the proposed model, our previous work, that combines the LSTM and attention, together with the MLP and LSTM model alone. The most relevant factor obtained in this paper is AC2 as input to the training model. The resulting MAPE results are shown in Table 2. The resulting RMSE results are shown in Table 3. The resulting MAE results are shown in Table 4.

In the case of less training data, using the clustering phase and prediction phase mentioned in this paper will reduce the prediction accuracy. When the time interval is 30 min, the amount of data is reduced to a quarter of the data volume with a time interval of 7.5 min. From the comparison of different time intervals, when the interval is 30 min, the prediction accuracy drops. On the contrary, in the case of more training data, the prediction accuracy can be effectively improved.

The experiment randomly selects two days of the month as a display of the predicted results. As shown in Figures 8–10, according to the law of PV power generation, the power generation in the early evening to the early morning of the next day is 0, which is not shown in Figures 8–10.

In Figures 8–10, (a-1) and (a-2), (b-1) and (b-2), and (c-1) and (c-2) are forecasting results of two days in winter; (a-3) and (a-4), (b-3) and (b-4), and (c-3) and (c-4) are forecasting results of two days in spring; (a-5) and (a-6), (b-5) and (b-6), and (c-5) and (c-6) are forecasting results of two days in summer; (a-7) and (a-8), (b-7) and (b-8), and (c-7) and (c-8) are forecasting results for two days in autumn. For winter, in Figure 10, deviations of the predicted power curve are found for 30 min ahead predictions. Satisfactory prediction performances in the winter season for 7.5 min ahead and 15 min ahead forecasts are achieved according to Figures 8 and 9.

Through the experimental clustering, training, and prediction stage, the accuracy of the forecasting model has a significant outperformance over the compared models when the amount of training data is sufficient. The clustering stage divides the original PV data into four different clusters. And each clustered data is trained by a hybrid deep learning framework combining CNN and LSTM. The sophisticated deep learning framework obviously enhances the forecasting performance over singular methods, such as MLP and LSTM. The AM technique already shows a great influence of the forecasting results in the experiments with ALSTM. The performance improvement is further enlarged for the proposed methods. According to Tables 2–4, the proposed model has a good performance on the RMSE indicator. Compared with the RMSE indicator, the MAE indicator reflects the actual situation of the prediction value error. The proposed model has the better performance than other models in the MAE indicator for intervals within 7.5 min. It is noted that the proposed model has significantly higher accuracy in the prediction of intervals within 7.5 min and has no obviously high accuracy compared with the other individual model in the prediction of intervals within 15 min and 30 min.

4. Conclusion and Discussion

Photovoltaic power generation prediction is of great significance for maintaining grid security and coordinating resource utilization. In the era of big data, it is possible for AI-driven IoT technology to perform accurate solar energy generation forecasting based on historical solar energy data [24, 30–32]. This paper proposes a hybrid deep learning method based on weather categories. Unlike traditional models, this experiment includes correlation analysis and clustering calculation of data, which effectively improves the generalization ability of the model and improves the prediction accuracy. The training algorithm employs the CNN algorithm, which can extract the data features more effectively and get more potential information in the data. Secondly, the attention mechanism is applied to the LSTM model, focusing on the extracted important features. In the prediction stage, the month of test set is used to determine

the forecasting model, which improves the accuracy of the prediction. This forecasting model is superior to the traditional algorithm model in predicting performance, and it also shows outstanding results in predicting other data types.

The experimental results shown in Section 3 show that the proposed method outperforms the traditional time series data prediction methods, such as MLP, LSTM, and ALSTM for PV system energy generation forecasting. Three evaluation metrics are used to show the superior performance, including RMSE, MAPE, and MAE. According to the results collected in Tables 2–4, the proposed method achieves higher forecasting accuracy for short-term and very short-term forecasting, e.g., the 7.5 min advanced forecasting. For longer-term forecasting, such as time intervals of 15 and 30 min, the forecasting performance advantage decreases.

One of the future works of this study is to extend the existing work for more generalized datasets, i.e., achieving acceptable forecasting results for longer-term forecasting of the PV system energy generation. Another future work direction is to apply the proposed framework towards a broader range of time series data applications in other fields, such as air quality forecasting [14, 35] and energy consumption forecasting [36].

Data Availability

Access to data is restricted. The solar irradiance data used in this study is collected by a local company in Shaoxing, China. The data is confidential and is only available for the company's internal usage.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Hangxia Zhou and Yang Du performed the conceptualization and validation; Hangxia Zhou contributed to the methodology and investigation; Ke Yan helped in the formal analysis and project administration; Yang Du participated in the data curation and wrote and prepared the original draft; Qian Liu wrote, reviewed, and edited the manuscript and helped in the supervision. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was funded by the Public Welfare Research Project of Zhejiang Province, China, grant number LGF18F020017, and in part by the National Natural Science Foundation of China under grant number 61850410531.

References

[1] P. Nema, R. K. Nema, and S. Rangnekar, "A current and future state of art development of hybrid energy system using wind and PV-solar: a review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 8, pp. 2096–2103, 2009.

[2] E. Lorenz, T. Scheidsteger, J. Hurka, D. Heinemann, and C. Kurz, "Regional PV power prediction for improved grid integration," *Progress in Photovoltaics: Research and Applications*, vol. 19, no. 7, pp. 757–771, 2011.

[3] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[4] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *In 2019 IEEE 39th international conference on distributed computing systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[5] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, no. 8, pp. 5790–5798, 2021.

[6] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.

[7] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2020.

[8] M. Qiu, S. Y. Kung, and Q. Yang, "Editorial: IEEE transactions on sustainable computing, special issue on smart data and deep learning in sustainable computing," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 1–3, 2019.

[9] K. Yan, L. Liu, Y. Xiang, and Q. Jin, "Guest editorial: AI and machine learning solution cyber intelligence technologies: new methodologies and applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6626–6631, 2020.

[10] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection based on digital twinning for smart manufacturing in industrial CPS," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[11] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2016.

[12] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[13] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[14] N. Jin, Y. Zeng, K. Yan, and Z. Ji, "Multivariate air quality forecasting with nested LSTM neural network," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[15] P. F. Jiménez-Pérez and L. Mora-López, "Modeling and forecasting hourly global solar radiation using clustering and classification techniques," *Solar Energy*, vol. 135, pp. 682–691, 2016.

[16] H. T. Yang, C. M. Huang, Y. C. Huang, and Y. S. Pai, "A weather-based hybrid method for 1-day ahead hourly forecasting of PV power output," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 3, pp. 917–926, 2014.

[17] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586–600, 2000.

[18] S. Bashyal and G. K. Venayagamoorthy, "Recognition of facial expressions using Gabor wavelets and learning vector quantization," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 7, pp. 1056–1064, 2008.

[19] Y. Han, N. Wang, M. Ma, H. Zhou, S. Dai, and H. Zhu, "A PV power interval forecasting based on seasonal model and nonparametric estimation algorithm," *Solar Energy*, vol. 184, pp. 515–526, 2019.

[20] W. VanDeventer, E. Jamei, G. S. Thirunavukkarasu et al., "Short-term PV power forecasting using hybrid GASVM technique," *Renewable Energy*, vol. 140, pp. 367–379, 2019.

[21] M. Malvoni, M. G. De Giorgi, and P. M. Congedo, "Forecasting of PV power generation using weather input data-preprocessing techniques," *Energy Procedia*, vol. 126, pp. 651–658, 2017.

[22] S. Srivastava and S. Lessmann, "A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data," *Solar Energy*, vol. 162, pp. 232–247, 2018.

[23] X. Jian, H. Gu, and R. Wang, "A short-term photovoltaic power prediction model based on dual-channel CNN and LSTM," *Electric Power Science and Engineering*, vol. 35, no. 5, pp. 7–11, 2019.

[24] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, and Y. Du, "Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism," *IEEE Access*, vol. 7, pp. 78063–78074, 2019.

[25] X. Chen, Y. Du, E. Lim, H. Wen, K. Yan, and J. Kirtley, "Power ramp-rates of utility-scale PV systems under passing clouds: module- level emulation with cloud shadow modeling," *Applied Energy*, vol. 268, p. 114980, 2020.

[26] J. Chang, Y. Du, X. Chen, E. G. Lim, and K. Yan, "Forecasting based virtual inertia control of PV systems for islanded microgrid," in *In 2019 29th Australasian universities power engineering conference (AUPEC)*, pp. 1–6, Nadi, Fiji, 2019.

[27] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[28] N. Jin, J. Wu, X. Ma, K. Yan, and Y. Mo, "Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification," *IEEE Access*, vol. 8, pp. 77060–77072, 2020.

[29] K. Yan, X. Wang, Y. Du, N. Jin, H. Huang, and H. Zhou, "Multi-step short-term power consumption forecasting with a hybrid deep learning strategy," *Energies*, vol. 11, no. 11, p. 3089, 2018.

[30] D. L. Fernandes, R. Siqueira-Batista, A. P. Gomes et al., "Investigation of the visual attention role in clinical bioethics decision- making using machine learning algorithms," *Procedia Computer Science*, vol. 108, pp. 1165–1174, 2017.

[31] J. Campbell, H. B. Amor, H. M. Ang, and G. Fainekos, "Traffic light status detection using movement patterns of vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 283–288, Rio de Janeiro, Brazil, 2016.

[32] X. Zhou, W. Liang, I. Kevin, K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 246–257, 2018.

[33] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[34] K. Yan, A. Chong, and Y. Mo, "Generative adversarial network for fault detection diagnosis of chillers," *Building and Environment*, vol. 172, p. 106698, 2020.

[35] Z. Zhang, Y. Zeng, and K. Yan, "A hybrid deep learning technology for PM 2.5 air quality forecasting," *Environmental Science and Pollution Research*, pp. 1–14, 2021.

[36] K. Yan, W. Li, Z. Ji, M. Qi, and Y. Du, "A hybrid lstm neural network for energy consumption forecasting of individual households," *IEEE Access*, vol. 7, pp. 157633–157642, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9951607, 13 pages
https://doi.org/10.1155/2021/9951607

Research Article

A Hybridly Optimized LSTM-Based Data Flow Prediction Model for Dependable Online Ticketing

Chunmei Fan,[1] Jiansheng Zhu,[2] Haroon Elahi,[3] Lipeng Yang,[2] and Beibei Li[2]

[1]*Graduate Department, China Academy of Railway Sciences, Beijing 100081, China*
[2]*Institute of Computing Technologies, China Academy of Railway Sciences, Beijing 100081, China*
[3]*School of Computer Science, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Haroon Elahi; haroon.elahi@e.gzhu.edu.cn

Received 10 March 2021; Accepted 16 May 2021; Published 8 June 2021

Academic Editor: Xiao Zhang

Fifth-generation (5G) communication technologies and artificial intelligence enable the design and deployment of sophisticated solutions for enhanced user experience and superior network-based service delivery. However, the performance of the systems offering 5G-based services depends on various factors. In this paper, we consider the case of the online railway ticketing system in China that serves the needs of hundreds of millions of people daily. This system's online access rates vary over time, and fluctuations are experienced, affecting its overall dependability and service quality. We use long short-term memory network, particle swarm optimization, and differential evolution to construct DP-LSTM—a hybridly optimized model to predict network flow for dependable and quality-enhanced service delivery. We evaluate the proposed model using real data collected over six months from the "12306 online ticketing" system. We compare the performance of the proposed model with mainstream network traffic prediction models. We use mean absolute percentage error, mean absolute error, and root mean square error for performance evaluation. Experimental results show the superiority of the proposed model.

1. Introduction

Fifth-generation (5G) communication technologies and artificial intelligence (AI) enable the design and deployment of sophisticated solutions for enhanced user experience and superior mobile service delivery meeting diverse critical requirements [1–4]. However, the performance of 5G-based services is dependent on many factors [5, 6]. 5G-based mobile services face challenges such as transferring of high data rates, rapid response requirements, dynamic coupling and decoupling of new devices, and their remote configuration [7]. Consequently, the overall data traffic associated with the service delivery systems grows tremendously [8]. Likewise, while 5G infrastructures support high data throughput and AI-based knowledge driven data layers try to address corresponding resource allocation and optimization challenges, Internet is a limited resource and new situations can affect its performance (https://www.nytimes.com/2020/03/26/business/coronavirus-internet-traffic-speed.html), which can lead to deteriorated service delivery.

Here, we need to remember that the performance of Internet-based services is not entirely dependent on the telecommunication infrastructure, and the scalability of servers entertaining service requests also plays a critical role [9]. Network congestion can slow down the service. Unexpected fluctuations in service request rates and corresponding changes in Internet traffic can directly affect the availability of web-based systems, and a substantial change in traffic may even result in crashing various application services (https://techcrunch.com/2018/12/26/alexa-crashed-on-christmas-day/). Consequently, we see that special measures and strategic approaches are used for load balancing, enhanced stability, and detecting and mitigating malicious actors affecting the availability of Internet-based services and systems [9–11].

Moreover, the requirements and challenges of Internet-based services vary for different application scenarios [12–14]. As a result, recent research has focused on application-specific issues investigating underlying factors affecting the performance of corresponding Internet-based services and proposing different optimization methods for improved

availability and Quality of Service (QoS) [15, 16]. In this paper, we focus on Chinese railway system that serves the needs of hundreds of millions of people daily. A key component of the Chinese railway system is the "12306 ticketing system (https://www.highspeed.mtr.com.hk/en/ticket/buy-ticket-12306.html)" that serves as the main channel for passengers to check the availability of tickets and make online bookings. This system can experience occasional incoming traffic fluctuations [17]. Particularly, with the wide-scale availability of high-speed Internet over 5G infrastructures, the number of people seeking the services through "12306 ticketing system" has grown significantly and ensuring its scalability and dependability is a significant challenge in its own.

This research observes and predicts the Internet traffic generated by the incoming service requests in the "12306 ticketing system" to propose a model that enables scalability and resilience in the event of Internet traffic surges. The main goal of proposing this model is to minimize the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the 12306 ticketing system. Since, AI and data analytics can play an important role to control and efficiently operate networks and for efficient service delivery [18–21], we use AI-based data analytics to achieve our goals.

There service requests and related network access flows are encountered at successive time intervals. Therefore, it can be treated as time series data. In other words, the problem of predicting network access flow for the "12306 ticketing system" resembles traffic flow forecasting and stock forecasting [22–24]. Conventionally, there are three main approaches used to solve these problems [25, 26]: traditional algorithms such as exponential smoothing and autoregressive integrated moving average (ARIMA) [27, 28]; traditional machine learning algorithms such as support vector regression (SVR) [28], eXtreme Gradient Boosting (XGBoost) [29, 30], and Random Forest; and deep learning algorithms, such as Deep Autoregressive Networks (DARNs) [31] and long short-term memory (LSTM) [32, 33]. Among these algorithms, the ARIMA requires stable data. It predicts network traffic flows considering the variations in the historical data. Therefore, it cannot predict nonlinear patterns, and its generalization ability is weak [34, 35].

Likewise, due to network flow's complex nature and the lack of typical behavior [36], traditional models cannot handle such data well. The LSTM network [32] has achieved good results in nonperiodic event detection [37], traffic load balancing [38], and other fields [8, 39]. It has shown promising performance in time series data trend prediction [34]. With its nonlinear approximation function and self-learning and self-adaptive features, LSTM can better describe the characteristics of time series data and achieve high prediction accuracy and strong generalization [33]. It is mainly used to describe the relationship between current data and previous input data and uses its memory ability to save the state information before the data is fed in the network and use the previous state information to influence the exact value and development trend of subsequent data. In LSTM, appropriate number of layers of and the number of hidden neurons in the feedforward network layer play key role in its performance. Research shows that increasing the number of network layers does not necessarily improve the effect, and selecting the appropriate network layers can train a highly accurate model [40, 41].

However, in the actual application of LSTM network, determining the network structure and parameter selection are challenging tasks. It is generally achieved by hit and trial method or based on experience. This is also a bottleneck in the development of neural network [41]. Particle swarm optimization (PSO) is a heuristic random search algorithm, which has a lower number of setting parameters, no update and mutation involved, and it can find the extreme function values faster. Some LSTM models use particle swarm optimization (PSO) algorithm to find the optimal super parameters and achieve good results [34, 38, 39]. However, the PSO algorithm converges faster in the early stage of the optimization process and is easy to fall into the local optimum in the later stage. In order to be able to establish an optimal model, it is proposed to realize the parameter selection of the LSTM traffic prediction model by fusing the particle swarm optimization (PSO) and the differential evolution (DE) algorithm [42]. Using DE to optimize the evolution of PSO can improve the results of PSO [43] and greatly reduce the probability of obtaining a local optimal solution.

In this paper, we use LSTM—optimizing it through a fusion of PSO and DE to construct the DP-LSTM model that we use for the access flow prediction of 12306 ticketing system. The purpose of proposing this model is to minimize the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the 12306 ticketing system.

Specifically, the contributions of this research can be listed as follows.

(1) We focus on "the 12306 ticketing system" to learn its flow data patterns and predict future traffic to minimize the impact of sudden fluctuations for improving its dependability and overall QoS

(2) We use long short-term memory (LSTM) network and hybrid optimization algorithm to construct the DP-LSTM model to predict network access traffic

(3) We evaluate the proposed model using real data collected over six months from the "12306 ticketing system" and compare its performance with mainstream time series data forecasting methods. We use mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE) for performance evaluation. Experimental results show the superiority of the proposed model over the benchmarks

2. Access Flow Data in "12306 Ticketing System"

As mentioned in the previous section, service requests and related network access flows are encountered at successive time intervals. Such data exhibits secular trend, cyclical fluctuations, and irregular variations [44]. Therefore, it can be expected that operations of the "12306 ticketing system"

can be affected by the time-varying cyclic and irregular variations. The system serves from 7 to 23, and we focused on the hourly peak flows and collected the data from March 1, 2020, to August 31, 2020. We conducted an exploratory analysis on the collected data to identify timing sequence and outliers. The results of this analysis are provided underneath.

2.1. Timing Sequence. By selecting the one-month data from March 1st, 2020, to April 1st, 2020, and observing the variation law of request flow, the timing sequence diagram obtained is shown in Figure 1. The timing diagram indicates that the flow rate fluctuations had evident periodicity and trend. It is basically in accord with the Internet ticketing system's daily operation schedule of suspending ticket booking from 0 : 00 am to 5 : 00 am and opening from 6 : 00 am to 11 : 00 pm. On March 5th, the flow showed an upward trend, because the tickets can be prebooked for April 4th, the Tomb Sweeping Day holiday. Every morning, the Internet ticketing system becomes available at 6 o'clock for ticket booking, resulting in a daily peak at 6 am. The stability of access flow data is further tested through autocorrelogram and unit root detection. The P value of the unit root check was 0.91. From the autocorrelogram shown in Figure 2, it can be found that it is without truncation. Hence, the request flow is not a stationary sequence.

2.2. Outlier Detection. We detected abnormal points of the data flow through the boxplots. The flows during the ticketing period (6 : 00 am-11 : 00 pm) and the nonticketing period (0 : 00-5 : 00am) were quite different. Therefore, we divided the data into the ticketing period and the nonticketing period to be tested. The test results are shown in Figure 3.

These red points represent abnormal data which are greater than Q3+2IQR. Q1 is the first quantile, Q3 is the third quantile, and IQR = Q3 − Q1. The abnormal data during the nonticketing period were the data at 5 : 00 am every day, which were the normal flow fluctuation before the ticket release. The abnormal data during the ticketing period appeared in the ticket prebooking for the Qingming Festival (from March 4 to March 9) and 6 : 00 am. The data were also normal business traffic fluctuations. Through outlier detection, there is no extreme abnormal data in the collected data. There is a big difference in traffic during ticketing and nonticketing periods, so the prediction models need to be built separately.

The analysis results indicated that the access flow of the Internet ticketing system was not only unsteady and periodic but also affected by holidays and ticket release timing, which was random and complicated. For the traditional timing sequence models, it is difficult for them to fit such time sequence data.

3. High-Level Design of DP-LSTM

In this section, we introduce different components that we use to construct the DP-LSTM model for predicting flow data for the "12306 ticketing system."

3.1. LSTM. The LSTM is a recurrent neural network (RNN), which effectively learns the long-term dependency relationship with well-designed "gated" architectures. It consists of memory cells and gate units. An LSTM neuron has input (multiplicative input), output (multiplicative output), and a forget gate. As the name suggests, the input gate handles input data stored in a given memory cell and protects it from perturbation by other irrelevant inputs. The output gate contains the output representations, and the forget gate handles the retention of historical information or, in other words, decides when to forget retained historical information. A typical LSTM network can have at least one input layer, one output layer, and a hidden layer. Memory cells and gate units are located in the hidden layer. When using LSTM, determining the network structure is a challenge, and it is often based on experience. Since single-layer LSTM is limited by the number of convolution kernels, multilayer LSTM can be a better choice [34]. Therefore, we use a multilayer LSTM in the network structure of the access flow prediction model of the railway ticketing system. At the same time, the dropout layer is added to improve the generalization ability of the model. Moreover, different studies have used different optimization algorithms to find the optimal LSTM prediction model [45–48]. We use a fusion of particle swarm optimization (PSO) and differential evolution (DE). The basic network structure is shown in Figure 4.

3.2. Particle Swarm Optimization (PSO). The PSO is a population-based stochastic algorithm for optimization. It searches for the optimal zone through the continuous interactions of population members (data), leading to an iterative improvement in the algorithm's performance [49]. It moves on its own in the search space and tests different parameters. It uses the group's optimal fitness to change the direction and distance of movement to complete the global search space's optimization process. Consider a group $X = \{x_1, x_2, \cdots, x_n\}$, consisting of m particles in a d-dimensional search space, and $x_i = [x_{i1}, x_{i2}, \cdots, x_{id}]$ is a group of training parameters of the DP-LSTM model. If x_i is featured with location $X_i^t = [x_{i1}^t, x_{i2}^t, \cdots, x_{id}^t]^T$ and velocity $V_i^t = [v_{i1}^t, v_{i2}^t, \cdots, v_{id}^t]^T$ individual optimal location: $p_i^t = [p_{i1}^t, p_{i2}^t, \cdots, p_{id}^t]^T$ and global optimal location: $p_g^t = [p_{g1}^t, p_{g2}^t, \cdots, p_{gd}^t]^T$ at the time t, the global-optimal location is the optimal parameter combination of the current training model; then, velocity and location of the particle at the time $t + 1$ can be updated to

$$\begin{cases} v_{id}^{t+1} = w v_{id}^t + c_1 r_1^t \left(p_{id}^t - x_{id}^t\right) + c_2 r_2^t \left(p_{gd}^t - x_{id}^t\right), \\ x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \end{cases} \tag{1}$$

where w is inertia weight which controls the effective equilibrium between global detection and local mining of the particle; c_1 and c_2 are learning factors which, respectively, adjust the step size flying to its own and global-optimal location; r_1 and r_2 are random numbers uniformly distributed within [0,1].

While PSO has its advantages, there is a possibility that after iteratively changing the optimal parameter combination with Equation (1), subsequent particle update is stuck into local

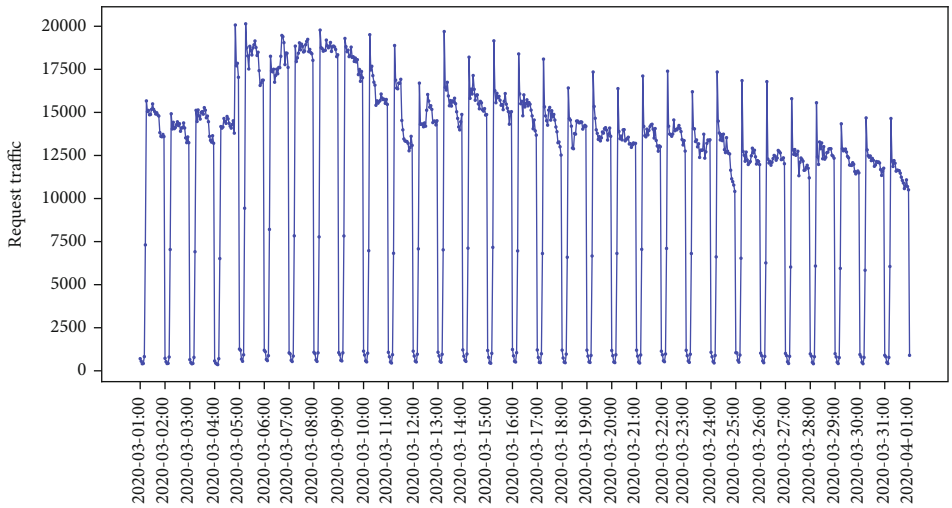


Figure 1: The timing sequence diagram.

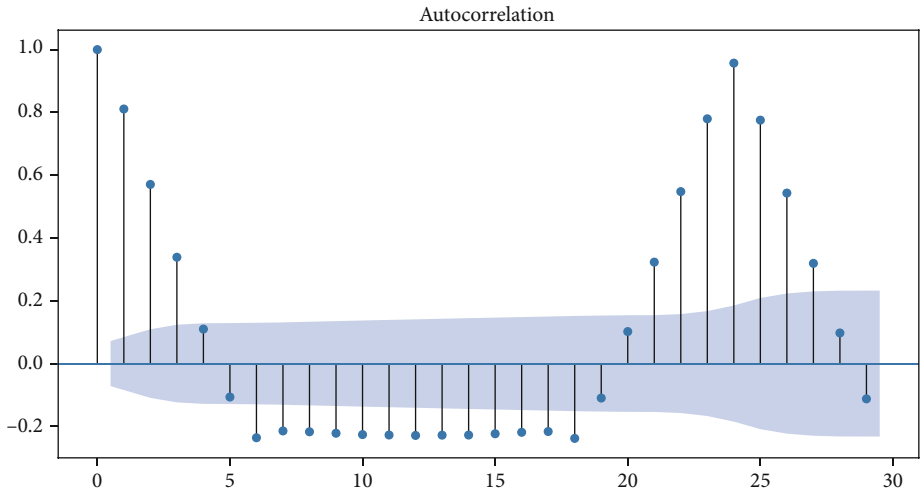


Figure 2: The autocorrelogram.

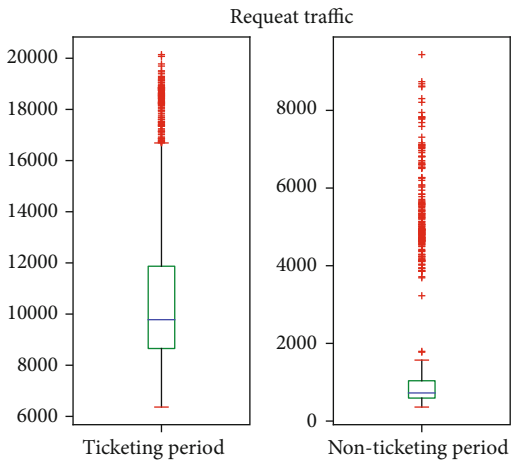


Figure 3: The boxplots of ticketing and nonticketing period data flows.

optimum. A differential evolution (DE) algorithm [20, 21] can be used to optimize the location update of the particle swarm with the optimal individual from the differential evolution swarm. In Equation (1), $p^t{}_{gd}$ uses the optimal value among differential individuals and particle swarm. Sharing the global-optimum of two swarms can accelerate optimization velocity of the particle swarm, reduce the risk of falling into local-optimum, and output the optimal parameter combination.

3.3. Differential Evolution. The DE algorithm is a parallel direct search method utilizing NP d-dimensional parameter vectors for optimization [42]. It is a simple yet effective technique based on group-random search, designed to solve the global optimization problem. Initially, the search vector population is randomly chosen, and it should cover the whole parameter space. New parameters are generated through the mutation operation that adds the differential weights of two parameter vectors to a third vector called a mutated vector. Mutated vector parameters "crossover" with those in a vector called target vector. The resulting vector is called a trial vector. Suppose the value of

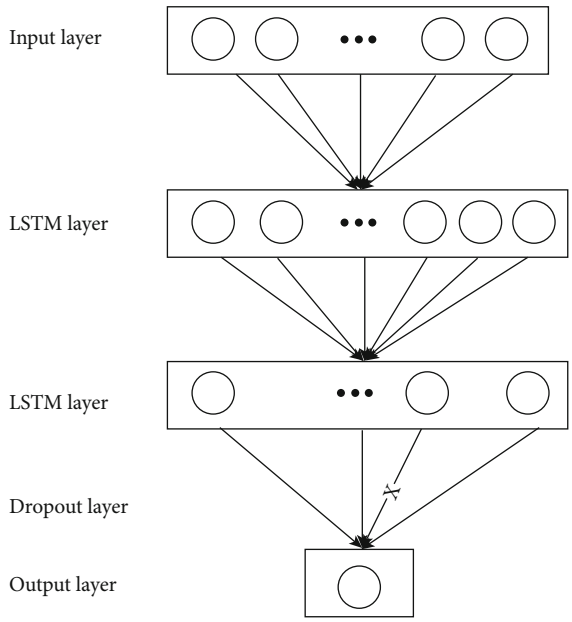


Figure 4: The fundamental network structure.

the trial vector's cost function is lower than the target vector. In that case, the trial vector replaces the target vector in the next generation. This process of replacing the target vector with the trial vector after comparing the cost function's value is called selection. The evolution of DE generates new descendants from the parental parameter vectors through mutation, crossover, and selection operation. In our design, the steps of DE evolution are as follows.

(1) *Swarm Initialization*. The initial swarm consists of a vector $P_i(t) = [P_{i.1}(t), P_{i.2}(t), \cdots P_{i.d}(t)]$ that consists of N parameters, which are randomly selected from the overall search space. d is the dimension of individual vector; $i \in \{1, 2, \cdots N\}$ present the i^{th} chromosome.

(2) *Mutation Operation*. Different strategies can be used for mutation operation. In order to share current optimal parameter combination with the particle swarm and accelerate the optimization velocity, Equation (2) represents an individual mutation operation.

$$v_i(t+1) = P_{\text{best}}(t) + F^* \left(P_{r_1}(t) - P_{r_2}(t)\right), \tag{2}$$

where i is the serial number of the current swarm; r_1 and r_2 are two unequal numbers randomly selected from $1 - N$ ($r_1 \neq r_2 \neq i$); $P_{\text{best}}(t)$ is the optimal individual in the t-generation particle swarm and differential swarm; F is the scaling factor.

(3) *Crossover Operation*. For the trial vectors generated from mutated and target vectors, Equation (3) is used for crossover selection to get the trial vector.

$$u_{i,j}(t+1) = \begin{cases} v_{i,j}(t+1), \text{if rand } (0,1) < \text{CR or rand } (j) = j, \\ P_{i,j}(t), \text{otherwise}, \end{cases} \tag{3}$$

where CR is the crossover probability, rand $(0, 1)$ is a random number uniformly distributed within [0,1], and rand (j) is a randomly selected dimension.

(4) *Selection Operation*. The parameter vector with high fitness (low-cost function value) is selected using Equation (4). This helps to select the optimal parameters for the next-generation swarm.

$$P_i(t+1) = \begin{cases} u_i(t+1), \text{if } f(u_i(t+1)) < f(P_i(t)), \\ P_i(t), \text{otherwise}. \end{cases} \tag{4}$$

The mixed optimization is to select the optimal value for $p^t{}_{gd}$ in Formula (1) and $P_{\text{best}}(t)$ in Formula (2) after iteration. The training here is to find the minimum loss value. Therefore, the optimal individual is selected with min $(f(p^t{}_{gd}))$, $f(P_{\text{best}}(t))$ as the basis of particle swarm and differential swarm during the next-generation evolution.

4. Implementation Details

In this section, we present the implementation details for the proposed DP-LSTM model. The model is constructed using LSTM—optimizing it through a fusion of PSO and DE. We use this model for the access flow prediction of the "12306 ticketing system." The purpose of proposing this model is to minimize the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the system. As mentioned in the previous sections, the flow data under consideration is time recurrent. We select several time points during ticketing and nonticketing periods. Figure 5 shows the flow trend. A cyclic flow variation can be noted with a significant difference in the access traffic flow depending on whether the ticketing is open or not. Therefore, we train different models for the ticketing period and nonticketing period. The overall structure of the network used for training this model is shown in Figure 4.

4.1. Input and Output. The real flow data for the past 24 hours is used as the model input. This is because the change period of the flow data is 24 hours. If i is the current hour, then the corresponding flow peak data within $[i - 24, i - 23, \cdots i - 3, i - 2, i - 1]$ hours is input, and the model will output the prediction for the peak traffic flow for the next hour. We also tried training the model using input comprising the flow data for more than 24 hours (multiple days). For example, Figure 6 shows some results generated by using traffic flow data for the past seven. We discovered that using long-term data for forecasting can better predict the trend of traffic changes. However, it introduces a serious lag and low prediction accuracy. Simultaneously, the increase in the number of days increases the input dimensions by seven

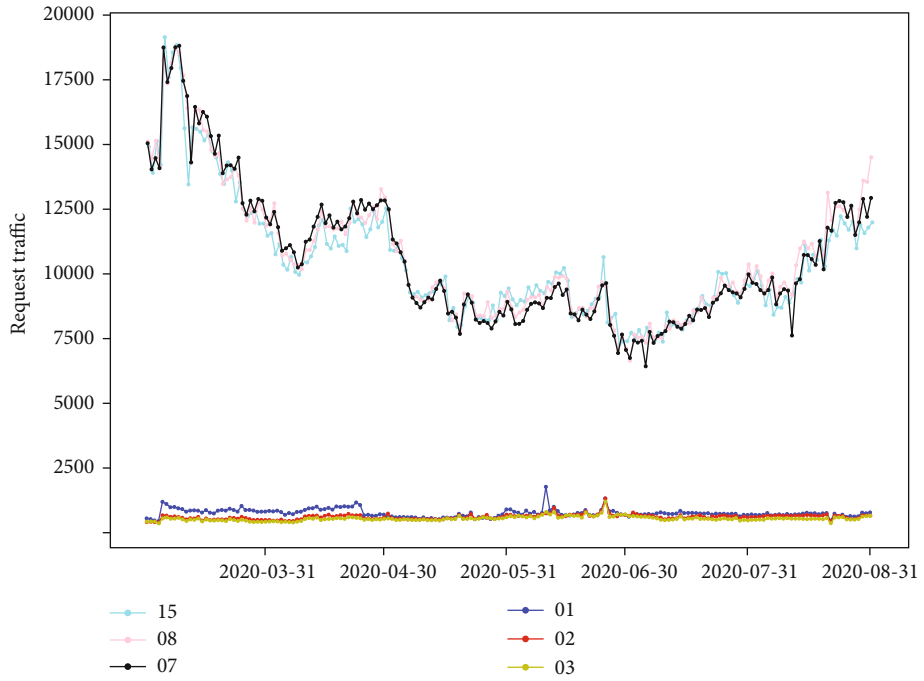


Figure 5: Service request flow variations at the same time, but on different dates.

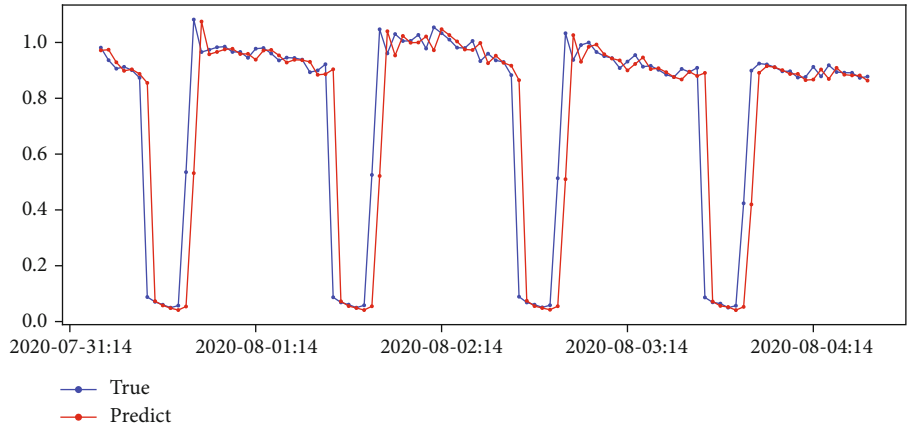


Figure 6: Model prediction results.

Table 1: Basic parameters.

Parameter	Description
NP_{PSO}	The number of particles
dim	Individual dimension
NP_{DE}	Population number
$iter_{max}$	Maximum number of iterations
w	Inertia weight of particles
CR	Crossover probability
c_1, c_2	Learning factors
F	Scaling factor

times. This significantly increased the time cost of model training. Therefore, selecting longer period data to predict the access traffic in the next hour is costly and does not generate optimal results.

4.2. Strategy Setting for Model Optimization

(a) Fitness function

The MAE between forecasted flow and real flow is considered as the fitness indicator. According to the experiment, the trained model's verified loss value has an inevitable fluctuation but tends to decline as a whole. Therefore, the average value of the final three verified loss values during training is used as a fitness measure.

(b) Boundary strategy

Table 2: Fitness of different models.

Iterations	PSO-LSTM (N)	DP-LSTM (N)
0	0.0117	0.0116
1	0.0113	0.0116
2	0.0109	0.0113
3	0.0108	0.0112
4	0.0108	0.0108
5	0.0106	0.0107
6	0.0106	0.0104
7	0.0106	0.0104
8	0.0106	0.0104
9	0.0106	0.0103
10	0.0106	0.0103

Table 3: Evaluation index results.

Model	MAE	MAPE	RMSE
LightGBM (N)	0.0193	8.5326	0.0480
RN-LSTM (N)	0.0191	10.9957	0.0391
PSO-LSTM (N)	0.0122	7.4350	0.0277
DP-LSTM (N)	0.0116	6.9252	0.0272

The boundary treatment is performed at each dimension of an individual observation, and maximum and minimum limits are set in each dimension. If an individual variation exceeds maximum or minimum, corresponding treatment is required. Equation (5) shows the treatment function.

$$\begin{cases} x_j = x_j^H \left(\text{if } x_j > x_j^H, 0 \le j \le n \right), \\ x_j = x_j^L \left(\text{if } x_j < x_j^L, 0 \le j \le n \right), \end{cases} \tag{5}$$

where j is the dimension of an individual variation, x_j^H, x_j^L is the maximum and minimum value of j, respectively, and n is the number of dimensions.

4.3. Hybrid Optimization Algorithm Flow. We fuse the evolution processes of PSO and DE to optimize the parameters of the model. This fusion helps identify the flow prediction model's optimal network structure during the ticketing and nonticketing period. The optimal parameters learned in this process are used for training to obtain the final flow prediction model. Underneath, we describe the detailed steps of algorithm flow.

Step 1: construct the sample sets for the nonticketing period and ticketing period, respectively. Generate the training sample set, validation sample set, and test sample set according to the ratio of 6 : 2 : 2. Then, perform model training for the nonticketing period flow prediction model DP-LSTM (N).

Step 2: initialize the basic PSO and DE parameters (as given in Table 1), and obtain the initial population. Generate a particle/individual randomly. Then, generate a corresponding flow prediction model according to the value of the particle, and calculate the fitness value of each model is using the fitness function. Finally, repeat the above operation to generate NP_{DE} individuals and NP_{PSO} particles.

Step 3: update the speed and position of parameters with the number of NP_{PSO} and record the optimal fitness value experienced by each particle using Equation (1). Also, update the overall optimal fitness value and optimal particle position of the PSO. Furthermore, perform the mutation, hybridization, and selection operations using Equations (2)-(4) on all individuals in the DE population. And update the overall optimal fitness value and optimal individuals of the DE population.

Step 4: select the overall optimal set of DE and PSO output as the evolutionary basis for the next epoch. If the optimal fitness value of the DE output is less than the optimal fitness value of PSO, the optimal fitness value and particle position of PSO should be updated and vice versa. Otherwise, the optimal value and individual corresponding to DE should be updated.

Step 5: test whether the algorithm reaches the maximum number of iterations $iter_{max}$ or the loss value is less than los s_{min}. If yes, obtain the optimal parameter combination and execute step 6; if not, update the iteration counter $t = t + 1$ and execute step 3.

Step 6: perform training according to the optimal parameter combination to obtain the optimal nonticketing period flow prediction model DP-LSTM (N).

Step 7: use the flow sample data set during the ticketing period to repeat step 2-step 6, obtain the optimal parameter combination model DP-LSTM (S) of the ticketing period, and combine the two optimal models to be the final flow prediction model DP-LSTM.

5. Experimental Evaluation

In this section, we provide the details of the experimental evaluation of the DP-LSTM model constructed to predict the flow data in the "12306 ticketing system" for warning and minimizing the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the system. We describe the experimental setup's details. We explain the data preprocessing procedures, measure the effect of fused optimization used in our design, compare the proposed model's performance with mainstream methods, and present the results of error analysis.

5.1. Experimental Setup. The proposed model was implemented using Python (3.7.3) and the TensorFlow2 framework. The experiments were performed on a 40-core machine with 125 Giga byte memory. Operating system is Red Hat 7.4. Python API Matplotlib (3.0.3) was used for generating plots.

5.2. Data Set. We collected real data from the "12306 ticketing system" for this research. The peak flow per hour from 1 March to 31 August 2020 (i.e., data of 4393 samples in total) was used to generate the sample set. The data set is a

Table 4: Important model parameters.

Model	Important parameters
LightGBM	Boosting type: GBDT, evaluation metric: RMSE, learning rate: 0.3, min child weight: 3.0, number of iterations: 152
XGBoost	Booster: GBTree, evaluation metric: RMSE, gamma: 0.55, max depth: 19, min child weight: 1.0, number of estimators: 26.0
Random forest	Bootstrap: true, max depth: 5, min sample leaf: 2, min sample split:2, number of estimators: 30
SVR	Kernel: RBF, C: 1, epsilon: 0.1
SARIMA	Autoregressive model (P), difference (d), moving average (q): (2, 0, 24)

Table 5: Evaluation index results.

Model	MAE	MAPE	RMSE
SARIMA	0.039	5.169	0.057
SVR	0.038	23.431	0.049
Random forest	0.029	4.376	0.047
XGBoost	0.030	7.970	0.043
LightGBM	0.024	4.329	0.035
DP-LSTM	0.021	3.483	0.033

time series data with a period of 24 hours. Through the analysis of "access flow data in 12306 ticketing system," it can be seen that the collected traffic belongs to nonstationary time series, and the change of access traffic has great fluctuation due to the influence of holidays and business changes. The data set was divided as per the "training set : verification set : test set = 6 : 2 : 2." The training set comprised the data collected from March to June 2020. The verification and test data set comprised the data collected in June and August 2020.

5.3. Performance Measures. We used mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE) to measure the model's performance. MAPE provides an effective determination method for forecasting the accuracy rate, which is usually expressed with percentage. The smaller the value is, the better the effect will be, as shown in Equation (6). The index MAE directly provides an average deviation between model output and real data. The larger the error is, the larger this value will be, as shown in Equation (7). RMSE is the standard deviation of the forecasting value, as shown in Equation (8). The smaller the value is, the better the model performance will be. MAPE, MAE, and RMSE are widely used evaluation indexes for neural network-based models [50].

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y}_i - y_i}{y_i}\right|, \tag{6}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y}_i - y_i|, \tag{7}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y\wedge_i - y_i)^2}, \tag{8}$$

where y_i is the real value and $\widehat{y}_i$ the predicted value.

5.4. Data Preprocessing. The internal covariate shift or change in the distribution of input variables used for training and testing the model obstructs a neural network-based model's training due to nonlinearities. Applying normalization on input data can offer an easy starting condition for the training and reduce the overall training time [51]. It is particularly the case for data sets collected over an extended period of time. Normalizing input data also reduces the risk of overfitting [52]. We test the input data to apply a suitable normalization technique. The Kolmogorov–Smirnov test (K-S test) is conducted on the collected data with the output result (statistic = 0.156 and P value = 6.31e-94), and the P value is far lower than 0.05. Therefore, the data cannot accommodate normal distribution. We cannot use the normalization based on mean variation for the sample data. Moreover, the maximum and minimum flows during different periods are different, and the training set's extreme values cannot be used for future data. Therefore, the normalization of maximum and minimum does not apply to the training data. The data distribution shows that the data is distributed within the scope of $[0, 2]^*$le7. Therefore, normalization is conducted by dividing by le7.

5.5. Impact of Hybrid Optimization. We built three models: RN-LSTM (N), PSO-LSTM (N), and DP-LSTM (N) for forecasting flow data during the nonticketing period to measure the impact of fusing PSO and DE. It is important to test that randomness does not play any role in the performance of a deep learning model [53, 54]. RN-LSTM (N) was constructed to serve this need. All RN-LSTM (N) parameters were set based randomly. The PSO-LSTM (N) uses the particle swarm optimization algorithm to select parameters for the LSTM model. The DP-LSTM (N) combines DE and PSO algorithms to select parameters for the LSTM model. The three models all use the network structure in Figure 4. The parameters to be optimized are the number of output units of LSTM in the first layer (out1) and the number of output units of LSTM in the second layer (out2), the number of training execution cycles (epoch), and the proportion of dropout (rate). The parameter sequence is [out1, out2, epoch, rate]. The boundary condition for PSO-LSTM (N) and DP-LSTM (N) is set to $L = [5, 5, 40, 0.01]$ and $R = [70, 30, 100, 0.4]$, respectively. Here, L and R refer to the left boundary and right boundary, respectively. The optimal parameter sequences determined by the RN-LSTM (N) model, the PSO-LSTM (N) model, and the DP-LSTM (N) model are [60,10,38,0.1], [41,14,81,0.02], and [70,11,97,0.01], respectively. The fitness values in each iteration of the PSO-LSTM (N) model and DP-LSTM (N) model are shown in Table 2. It can be seen

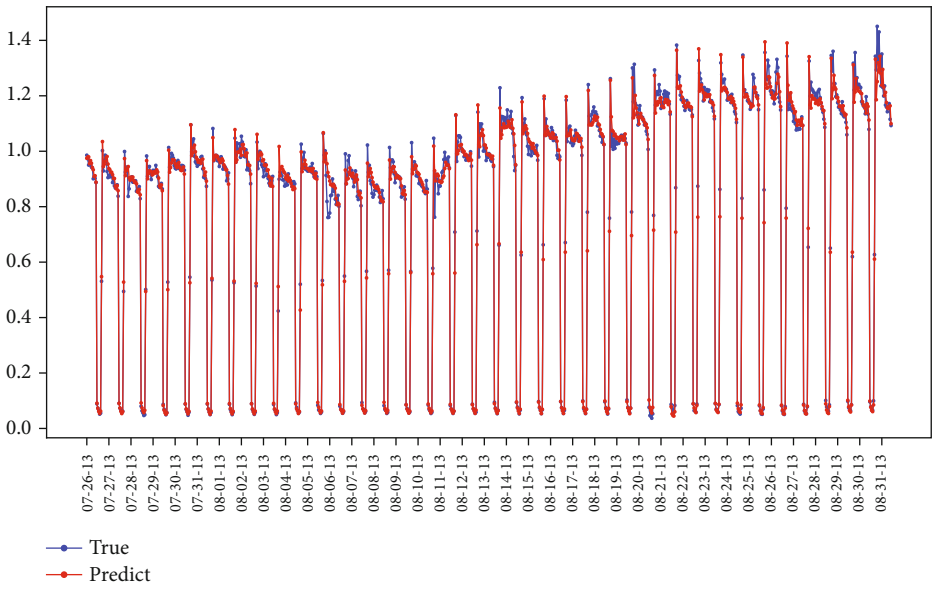


Figure 7: Overall forecasting effect.

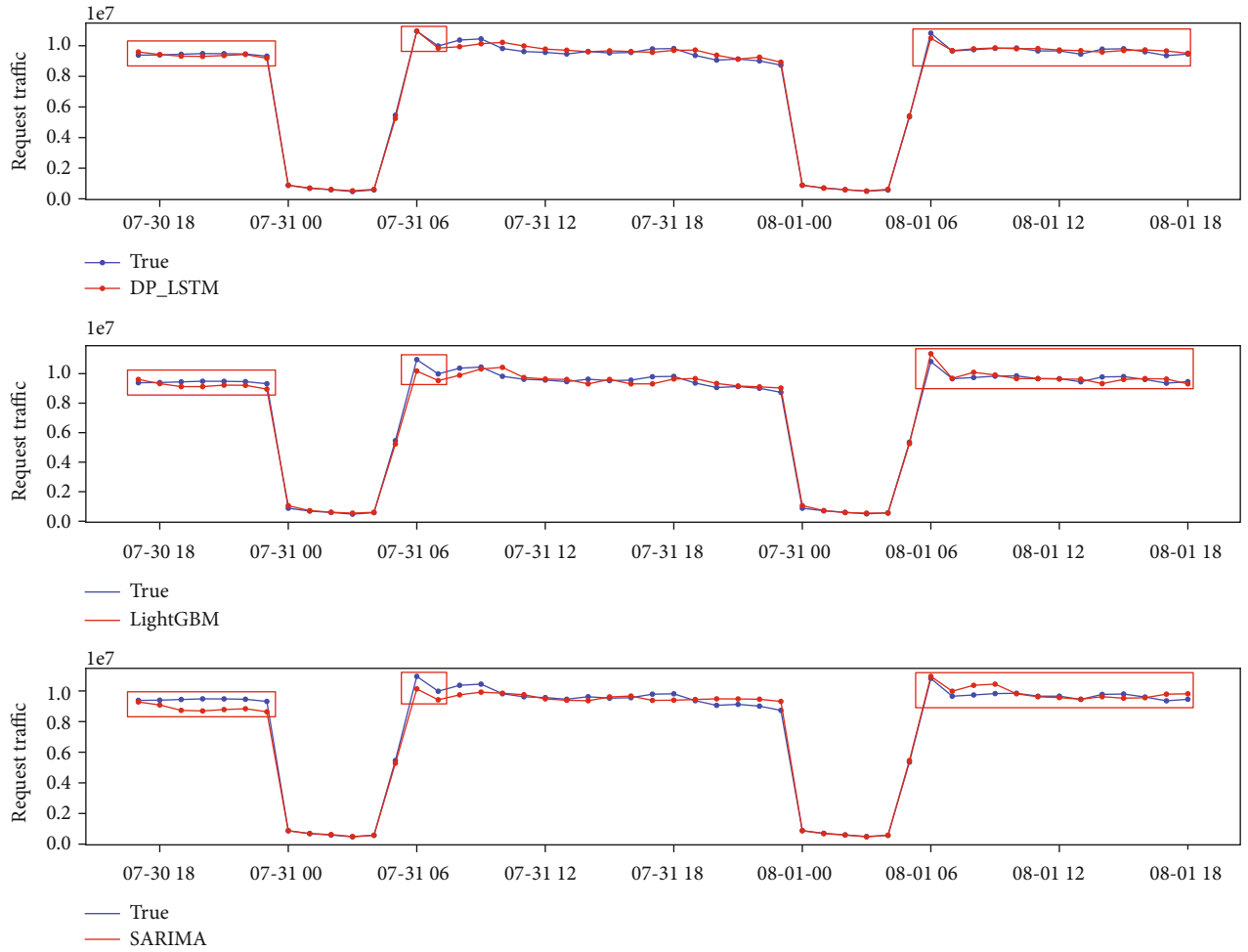


Figure 8: Prediction effect of different models.

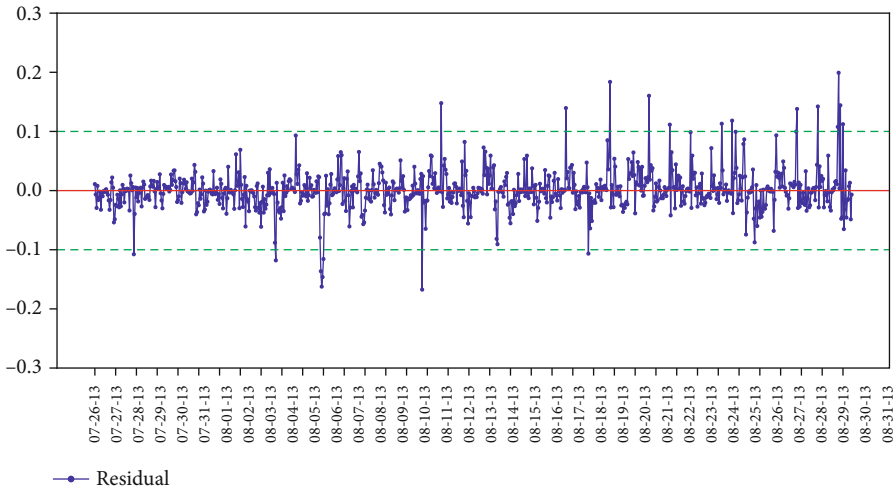


Figure 9: Residual sequence.

from the fifth iteration that PSO-LSTM (N) falls into the local optimum and cannot update the optimal value anymore. But PSO-LSTM (N) uses DE to guide the evolution of PSO, and the two populations develop and explore coordinately, which reduces the risk of falling into local optimum and speeds up the optimization.

To further compare parameter influence on the model, the optimal model LightGBM (N) was built using LightGBM for benchmarking. The test set was forecasted with RN-LSTM (N), PSO-LSTM (N), DP-LSTM (N), and LightGBM (N). The MAE, MAPE, and RMSE evaluation indexes were calculated, respectively, with the Formulas (5)–(7). Table 3 shows the results. It can be seen that MAE, MAPE, and RMSE of the DP-LSTM (N) model are lower than RN-LSTM (N) and PSO-LSTM (N). Therefore, it is possible to search optimal model parameters with mixed optimization based on DE and PSO. In contrast, the model with randomly selected parameters has worse effects for the same network structure, and some indexes are worse than LightGBM (N). Consequently, parameter optimization is crucial for the LSTM model results. The proposed mixed optimization method can achieve better effects.

5.6. Comparison with Mainstream Approach. We compare the DP-LSTM model with the mainstream time series models to verify the forecasting effect on the access flow of the "12306 ticketing system." The Light Gradient Boosting Machine (LightGBM), XGBoost, Random forest, support vector regression (SVR), and Seasonal Autoregressive Integrated Moving Average (SARIMA) algorithm are used to build models. These methods have been widely used in recent studies [55–59]. The parameter optimization is conducted for each algorithm model for building the optimal models. For example, LightGBM and XGBoost use Bayesian optimization for parameter optimization, and SARIMA uses Akaike Information Criterion (AIC) for parameter selection. Table 4 presents related parameters for different models.

Each model makes predictions for the same test set, and then forecasting results are compared. Table 5 presents the evaluation results of each model for given indexes. The DP-LSTM generated the best value for the MAPE index. Contrarily, SVR and SARIMA have the worst values. Each index of LightGBM in the machine learning algorithm is approximate to DP-LSTM.

Figure 7 shows the overall forecasting effect of DP-LSTM. It can be seen that the model can achieve better fitting for data fluctuations.

The forecasts generated using the LightGBM, SARIMA, and DP-LSTM are shown in Figure 8. Reviewing the evaluation indexes in Table 5 and Figure 8, it can be observed that the prediction effect of the DP-LSTM is better than traditional algorithm. DP-LSTM needs to train the structure and weight parameters. The training time of DP-LSTM is about 3 hours, which is longer than that of LightGBM, but shorter than that of SARIMA. The training speed of DP-LSTM needs to be further optimized.

5.7. Residual Sequence Analysis. Residual sequence analysis (error analysis) plays a crucial role in time series prediction and analysis. If a time series is a white noise, there is a zero correlation among this series' values. Prediction models do not work well on such series. Contrarily, while forecasting a time series, ideally, the series comprising a model's forecasting errors should be white noise. Suppose the forecast errors constitute a series that is not white noise. In that case, the predictive model can be further optimized. We get residual sequence by subtracting the forecasted value of DP-LSTM from the test set's real value. We use the white noise test function of Python for testing. The residual is not a white noise sequence because the P value is lower than 0.05 when the lag is between one and forty. Although the accuracy of the model training is high, there is extractable information that shall be further optimized to improve the forecasting accuracy. Figure 9 shows the distribution of the residual sequence. Although there is related information in the residual sequence, the error changes within 0.1. Therefore, the error

must be considered when the model is applied for decision-making.

6. Conclusion

The 5G communication technologies and their AI-embedded infrastructures enable the design and deployment of sophisticated network-based services. However, the hosts deploying these services can experience dependability and QoS issues. This paper constructed a deep learning-based model DP-LSTM using LSTM and hybrid optimization to address such problems in the "12306 ticketing system." This system serves hundreds of millions of railway passengers daily and can experience network flow fluctuations due to demand variation. The proposed model forecasts the network flow data peak for the next hour based on the recent day's access data. The performance of the LSTM network structure is optimized through a fusion of DE and PSO optimization algorithms. We used MAPE, MAE, and RMSE to evaluate the proposed model's performance using real data experimentally. A comparison with the mainstream time series forecasting algorithms demonstrated the superiority of the proposed model. However, error analysis/residual sequence analysis showed that the proposed model could be further optimized. The proposed system can help for resource planning of the "12306 ticketing system," thereby improving its dependability and QoS. Such solutions can also reduce the overall costs, particularly in cloud-based environments driven by pay-per-use model.

Data Availability

Data can be obtained by contacting the first author (fan.chun.mei@163.com).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported in part by the Chinese Academy of Railway Sciences under Grant Number 2019YJ122 and in part by the National Key R&D Program of China under Grant Number 2019YFF0301400.

References

[1] R. Li, Z. Zhao, X. Zhou et al., "Intelligent 5G: when cellular networks meet artificial intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, 2017.

[2] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection based on digital twinning for smart manufacturing in industrial CPS," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[3] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.

[4] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Transactions on Services Computing*, vol. 13, pp. 1–1, 2019.

[5] C. N. Tadros, M. R. M. Rizk, and B. M. Mokhtar, "Software defined network-based management for enhanced 5G network services," *IEEE Access*, vol. 8, pp. 53997–54008, 2020.

[6] M. Shariat, Ö. Bulakci, A. de Domenico et al., "A flexible network architecture for 5G systems," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 5264012, 19 pages, 2019.

[7] S. Din, A. Paul, and A. Rehman, "5G-enabled hierarchical architecture for software-defined intelligent transportation system," *Computer Networks*, vol. 150, pp. 81–89, 2019.

[8] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.

[9] Y. Bai, P. Hao, and Y. Zhang, "A case for web service bandwidth reduction on mobile devices with edge-hosted personal services," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 657–665, Honolulu, HI, USA, 2018-April.

[10] S. Wang, H. Elahi, Y. Hu, Y. Zhang, and J. Wang, "A botnets control strategy based on variable forgetting rate of control commands," *Concurrency and Computation: Practice and Experience*, pp. 1–21, 2020.

[11] K. Wang, H. Li, Y. Feng, and G. Tian, "Big data analytics for system stability evaluation strategy in the energy internet," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1969–1978, 2017.

[12] K. S. Kim, D. K. Kim, C. B. Chae et al., "Ultrareliable and low-latency communication techniques for tactile internet services," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 376–393, 2019.

[13] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2020.

[14] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, pp. 1–1, 2016.

[15] J. Chen and J. Chen, "Stability analysis and parameters optimization of islanded microgrid with both ideal and dynamic constant power loads," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3263–3274, 2018.

[16] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, "A blockchain-based multi-cloud storage data auditing scheme to locate faults," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2021.

[17] F. Liu, Z. Sun, P. Zhang, Q. Peng, and Q. Qiao, "Analyzing capacity utilization and travel patterns of Chinese high-speed trains: an exploratory data mining approach," *Journal of Advanced Transportation*, vol. 2018, Article ID 3985302, 9 pages, 2018.

[18] D. D. Clark, C. Partridge, J. Christopher Ramming, and J. T. Wroclawski, "A knowledge plane for the internet," *Computer Communication Review*, vol. 33, no. 4, pp. 3–10, 2003.

[19] A. Mestres, A. Rodriguez-Natal, J. Carner et al., "Knowledge-defined networking," *Computer Communication Review*, vol. 47, no. 3, pp. 2–10, 2017.

[20] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[21] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A blockchain-enabled deduplicatable data auditing mechanism for network storage services," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2020.

[22] J. Li and J. Wang, "Short term traffic flow prediction based on deep learning," in *CICTP 2019*, pp. 2457–2469, American Society of Civil Engineers, Reston, VA, 2019.

[23] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," *Procedia Computer Science*, vol. 167, no. 2019, pp. 2091–2100, 2020.

[24] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.

[25] B. Krollner, B. Vanstone, and G. Finnie, "Financial time series forecasting with machine learning techniques: a survey," *In Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN 2010): Computational Intelligence and Machine Learning*, pp. 25–30, 2010.

[26] G. Mahalakshmi, S. Sridevi, and S. Rajaram, "A survey on forecasting of time series data," in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pp. 1–8, Kovilpatti, India, 2016, IEEE.

[27] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, 2015.

[28] K. Lin, Q. Lin, C. Zhou, and J. Yao, "Time series prediction based on linear regression and SVR," in *Third International Conference on Natural Computation (ICNC 2007)*, pp. 688–691, Haikou, China, 2007.

[29] Y. Song, *Stock Trend Prediction: Based on Machine Learning Methods*, UCLA, 2018, https://escholarship.org/uc/item/0cp1x8th.

[30] S. Cerna, C. Guyeux, H. H. Arcolezi, R. Couturier, and G. Royer, "A Comparison of LSTM and XGBoost for Predicting Firemen Interventions," in *Trends and Innovations in Information Systems and Technologies*, WorldCIST 2020, Á. Rocha, H. Adeli, L. Reis, S. Costanzo, I. Orovic, and F. Moreira, Eds., Springer, Cham, 2020.

[31] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra, "Deep autoregressive networks," *31st International Conference on Machine Learning*, vol. 4, pp. 2991–3000, 2014.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] M. Ragupathy and X. S. Ma, "Long short term memory based total traffic prediction for container load balancing," *Technical Disclosure Commons*, p. 4, 2018.

[34] M. Li, Y. Wang, Z. Wang, and H. Zheng, "A deep learning method based on an attention mechanism for wireless network traffic prediction," *Ad Hoc Networks*, vol. 107, p. 102258, 2020.

[35] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[36] M. Meiss, F. Menczer, and A. Vespignani, "On the lack of typical behavior in the global web traffic network," in *Proceedings of the 14th international conference on World Wide Web - WWW'05*, pp. 510–518, New York, New York, USA, 2005, ACM Press.

[37] B. Song, C. Fan, Y. Wu, and J. Sun, "Data prediction for public events in professional domains based on improved RNN-LSTM," *Journal of Physics: Conference Series*, vol. 976, article 012007, 2018.

[38] T. Dlamini and S. Vilakati, "LSTM-based traffic load balancing and resource allocation for an edge system," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8825396, 15 pages, 2020.

[39] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019, IEEE.

[40] S. Liu, G. Liao, and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2787–2790, Wuhan, China, 2018, IEEE.

[41] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An efficient privacy-enhanced attribute-based access control mechanism," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 5, 2020.

[42] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[43] H. Fu, "A novel hybrid differential evolution and particle swarm optimization algorithm for binary CSPs," in *2012 International Conference on Computer Science and Electronics Engineering*, pp. 545–549, Hangzhou, China, 2012.

[44] Y. Dodge, *The Concise Encyclopedia of Statistics*, Springer New York, New York, NY, 2008.

[45] A. E. R. ElSaid, F. El Jamiy, J. Higgins, B. Wild, and T. Desell, "Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration," *Applied Soft Computing Journal*, vol. 73, pp. 969–991, 2018.

[46] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Computers and Industrial Engineering*, vol. 143, no. March, article 106435, 2020.

[47] K. Chen, "APSO-LSTM: an improved LSTM neural network model based on APSO algorithm," *Journal of Physics: Conference Series*, vol. 1651, article 012151, 2020.

[48] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, and J. Zhang, "Hierarchical bidirectional RNN for safety-enhanced B5G heterogeneous networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.

[49] J. Kennedy, "Particle swarm optimization," *Studies in Computational Intelligence*, vol. 780, pp. 760–766, 2019.

[50] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: the state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[51] J. Wang, S. Li, Z. An, X. Jiang, W. Qian, and S. Ji, "Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines," *Neurocomputing*, vol. 329, pp. 53–65, 2019.

[52] M. Liu, W. Wu, Z. Gu, Z. Yu, F. F. Qi, and Y. Li, "Deep learning based on batch normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

[53] X. Huang, G. C. Fox, S. Serebryakov, A. Mohan, P. Morkisz, and D. Dutta, "Benchmarking deep learning for time series: challenges and directions," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 5679–5682, Los Angeles, CA, USA, 2019.

[54] X. Zhou, W. Liang, K. I.-K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 246–257, 2019.

[55] Y. Wang, S. Sun, X. Chen et al., "International Journal of Electrical Power and Energy Systems Short-term load forecasting of industrial customers based on SVMD and XGBoost," *International Journal of Electrical Power & Energy Systems*, vol. 129, no. February, article 106830, 2021.

[56] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[57] C. Faloutsos, J. Gasthaus, T. Januschowski, and Y. Wang, "Forecasting big time series," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 2102–2105, 2018.

[58] P. F. Pai, K. P. Lin, C. S. Lin, and P. T. Chang, "Time series forecasting by a seasonal support vector regression model," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4261–4265, 2010.

[59] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "Blockchain-enabled accountability mechanism against information leakage in vertical industry services," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6657580, 10 pages
https://doi.org/10.1155/2021/6657580

Research Article

Construction of Trusted Routing Based on Trust Computation

Bei Gong, Jingxuan Zhu, and Yubo Wang

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Correspondence should be addressed to Yubo Wang; wangyubo@bjut.edu.cn

Received 24 December 2020; Revised 2 February 2021; Accepted 23 March 2021; Published 19 April 2021

Academic Editor: Xiao Zhang

In the field of applied IoT, a large number of wireless sensor devices are tasked with data production and collection, providing IoT subjects with a large amount of basic data to support top-level IoT applications. However, there is a considerable risk of being attacked on such sensor networks that are organized in a wireless form. These relatively independent network devices have extremely limited performance and lifetime, a problem that can be supplemented in a centralized network with base stations by relying on the performance of the core nodes of the network, but in a decentralized self-organizing network, they can have a serious adverse impact on the implementation of security solutions. Considering the fundamental nature of the data generated by such end devices in IoT application services, the protection of their security is also directly related to the quality of upper layer services provided. The main research result of this paper is the design of a trust routing scheme for self-organizing networks. The scheme is based on a comprehensive evaluation of data transmission rate, transmission delay, and other factors related to the operation status of the self-organized network and improves the efficiency of the overall work of the self-organized network by reducing the performance consumption of individual nodes of the self-organized network and balancing the network load.

1. Introduction

With the popularization and development of Internet technology, the old Internet infrastructure services gradually fail to meet the demand for Internet application services. As a form of network connection between network data and service devices, the deep connection to users and network devices and a wide range of application scenarios make Internet of Things (IoT) technology an important way to develop network services. And based on the development of mobile terminal technology, the field of IoT technology has gradually emerged as a new development direction represented by mobile Internet-built mobile IoT technology, with more emphasis on the flexibility of IoT devices in deployment, management, use, and other aspects. Therefore, mobile IoT technology has been developed by many Internet-developed countries as the next stage of important development goals and direction and to a large number of human and financial investment.

With the advancement of the technology and the application process, some new issues are emerging around mobile IoT. The dependence of the IoT on the Internet and the lack of basic security technologies make the overall security of the IoT a serious threat at the same time as the rapid development of the IoT application technology. This threat is reflected, on the one hand, in the fact that low-protection IoT terminal devices are more likely to be affected by network attacks, and on the other hand, when a large number of low-protection devices with network data transmission capabilities are controlled by a network attacker, the attacker can use this as a basis to further expand the threat of network attacks.

According to the predictions [1] made by Cisco, a leading provider of network equipment and services, regarding the data transmission during the operation of global networks, using 2018 statistics as a basis, Cisco believes that the number of data network users worldwide will increase 51% by 2023 from 3.9 billion in 2018, while there will be 29.3 billion network devices connected to the global Internet, in which the development of IoT technology has played an important role. In the forecast data, M2M devices [2], which are representative of IoT devices with services covering multiple

applications such as smart towns, smart grids, smart climate management, smart shopping, and smart picking, will account for more than 50% of the total network devices, making them a major component of global network connectivity. Such rapid network growth in the good direction will promote new concepts such as social IoT (SIoT) [3], which shares services over the Internet, while in the bad direction will lead to a rapid increase in attacks against network devices occurring in the network, with a 39% increase in DDoS attacks alone globally between 2018 and 2019 [1]. In 2016, the Mirai malware contaminated about 2.5 million endpoint devices and used them to carry out further network attacks [4]. A botnet malware attack called Dark Nexus, spanning China, Thailand, Brazil, South Korea, and Russia in a reverse proxy fashion, was also seen in April 2020 [5]. As a direct result of these problems, the applications built on the IoT infrastructure framework cannot guarantee the security and reliability of their application services. They also indirectly affect the reliable operation of other networks, as controlled IoT devices can be used to launch new network attacks. This risk poses a significant threat to IoT technologies, which rely on extensive data collection and remote interaction, with extremely high data transmission reliability requirements. This explosive growth accompanied by elevated industry risk requires effective countermeasures to be taken in order for mobile IoT application technology to achieve long-term growth [6].

In order to establish a secure and effective data transmission mechanism in the Internet of Things (IoT), which guarantees the security of data transmission while reducing the additional performance impact generated, this paper designs a wireless router establishment scheme based on distributed trust assessment. Firstly, a feasible data collection and trust assessment model is designed for the operating environment of the wireless self-organized network of IoT in order to realize the conversion from network state to quantitative trust assessment results. Then, based on the results of such a trust assessment, a set of route establishment schemes based on trust assessment is designed in combination with path length, transmission delay, and other metrics related to the network routing process. And a corresponding information maintenance mechanism to maintain and respond to network state changes is designed.

2. Related Work

Trust-based security is a new way of providing security without using cryptography approaches [7] and can effectively perform node anomaly behavior detection and overall network situational awareness tasks in an IoT environment. In terms of efficiency, avoiding or to some extent reducing the additional overhead in the execution of target devices for security through trusted security is also more suitable for use on IoT devices with limited resources. Trust in the field of wireless communication networks may be defined as the degree of reliability of other nodes performing actions [8, 9]. By evaluating this degree of reliability, it is possible to gain an understanding of the operation of each object in the network, and through the process of evaluating the trust, it is possible to obtain an assessment of the changes in the overall security situation of the network and to detect security events in the network. In the Internet of Things (IoT), which simultaneously has multiple characteristics of wireless transmission networks, social networks, and P2P networks [10], trust models are well suited to guide the overall operation of the network by describing its interactive behavior. For a self-organizing network at the end of the Internet of Things, the process can be described as some members acting socially to provide or request services from other members. A class of trust protocols such as [11–15] has emerged during research and development on self-organizing network security at the end of the Internet of Things (IoT). It is built on basic trusted security techniques, but the trust calculation process is complex and difficult to apply in sensor mobile self-organizing networks. And social network trust protocols such as [16–18] introduce social network relationships in trust management and enhance network security, but still fail to consider network operation efficiency well and lack countermeasures related to the spatial location change of sensing devices.

In terms of trusted route establishment, the security-aware ad hoc routing (SAR) [19] scheme designed by Sherchan et al. provides basic route discovery and route security for the route establishment problem in a wireless self-organizing network environment. It designed a node trust level management mechanism to evaluate the security of network nodes in the region. However, the complex computational process associated with secure schemes based on encryption and decryption algorithms can lead to degradation of internode transmission performance, which is unacceptable in sensor networks with relatively limited computational resources and energy.

The CENtralized Trust-based Efficient Routing (CENTERA) protocol [20] designed by Yi et al. uses authentication as the basis of trust and establishes a trust assessment for network nodes based on their forwarding situation. It performs the trust evaluation process and trust-based node state discrimination and malicious node quarantine through a base station designed in the solution. The trusted state of the node obtained from the forwarding case can be directly associated with the data forwarding process of routing, and the message and authentication computation, which is less expensive than the encryption and decryption computation, can also be used to protect the security of the message as it is propagated. However, the scheme's dependence on a base station prevents it from being directly applied to a self-organizing network of sensor devices. Also, such a centralized trust management model has a certain degree of lag in the feedback control process.

Based on existing research results related to trust models and trusted routing schemes, and taking into account the network situation of sensor self-organizing networks, this paper designs a routing scheme, including a self-organizing network trust assessment model, a trust-based routing scheme, and continuous maintenance of routing information. Compared with the above scheme, this proposal focuses on the application of trustworthy evaluation in distributed scenarios and the application of encryption algorithms in the network of IoT end devices

3. Trust Assessment Model

The design goal of the scheme is to address the problem of route security in sensor self-organizing networks. In the routing process, the core interaction is the transmission of data over the wireless medium, and the scheme divides this transmission process into two stages: route discovery and data relay. The route discovery stage is the network establishment stage, where devices exchange routing information several times to establish a basic data framework for the self-organizing network topology. In the data relay stage, the sensor dynamically carries out multiple relay node selection processes on multiple nonrepeating sensor devices according to the routing information established in the previous stage to complete the transmission of data from the source device to the destination device.

The trust assessment model in this scheme consists of three parts: trust model, trust data acquisition, and trust maintenance. The trust model describes the overall process of assessing the trustworthiness of a node, including a quantitative assessment of data related to the operational state of the network and a computational scheme for obtaining standard comprehensive trust assessment results based on the quantitative assessment. Trust data acquisition describes the way nodes acquire raw data in the trust assessment process, as well as the process of authenticating and assessing the trustworthiness of the data itself. Trust maintenance describes a scheme for maintaining the validity of a self-organized network trust state by controlling trust assessment behavior and responding to security events that occur during operation.

3.1. Trust Model. The design of the scheme's trust model is primarily used to describe the method by which nodes evaluate the process of data exchange between the network environments in which they are located, i.e., with other network nodes. In general, this evaluation process can be described as the evaluation and quantification of the data transmission rate, data transmission delay, and other parameters by a node in a self-organized network to calculate the comprehensive trust assessment value of other nodes in the same self-organized network.

In the specific design, the scheme uses a distributed trust assessment approach for the decentralized self-organized network of end devices. Suppose there are n network nodes p_1 to p_n in a self-organizing network M, a node p in a self-organizing network needs to autonomously complete a trust assessment of all other $n-1$ nodes, including the collection of network state data and the results of other node trust assessments.

3.1.1. Direct Trust Assessment. As part of the trust model, the direct trust assessment corresponds to the trust calculation based on the network state data described above; to complete this computational process, this paper designs a direct trust evaluation function $\mathrm{dta}(p)$ to quantify the network behavior of the target node. Suppose there are two network nodes A and B in a self-organizing network, and node A has to perform a direct trust assessment on node B. In this process, node A needs to obtain the following data: the list of network state parameters L_{ns}, the network state quantization evaluation function $\mathrm{qa}()$, and the network state evaluation weights $W_{\mathrm{ns}}.L_{\mathrm{ns}}[\mathrm{pno}, \mathrm{parno}]$ containing multidimensional operational state data collected by node A for other nodes. The function $\mathrm{qa}()$ describes the algorithm for quantitative evaluation of each dimension of data in $L_{\mathrm{ns}}.W_{\mathrm{ns}}[\mathrm{parno}]$ which contains the calculated weights of the results of the quantitative assessment of each state parameter in node A in the comprehensive trust assessment process. In summary, as an example of direct trust assessment of the m-dimension of node A on node B, the direct trust assessment function can be expressed as

$$\mathrm{DT} = \mathrm{dta}(p_{\mathrm{B}}) = \sum_{i=1}^{m} W_{\mathrm{ns}i}\mathrm{qa}\left(L_{\mathrm{ns}[\mathrm{B},i]}\right). \tag{1}$$

On this basis, it is only necessary to unify the dimensional composition of the network state parameter list and the network state quantization evaluation function within a self-organized network to obtain a relatively uniform standard of direct trust evaluation data to support the subsequent trust evaluation process.

3.1.2. Indirect Trust Assessment. In the model, the indirect trust assessments correspond to the trust calculations based on the results of other node trust assessments above. In the process of trust assessment of nodes, due to differences in the level of interaction and spatial distribution between nodes, in a relatively large self-organized network, the data obtained by a node through direct data collection and the corresponding direct trust assessment results are only effective in reflecting the actual operational status of a small number of nodes. For other (typically spatially distant) nodes, the results of the trust assessment need to be corrected by indirect data. This indirect data can be raw network state data collected by other nodes or direct trust assessment results computed by other nodes that have not been corrected for indirect trust. Using raw data avoids inconsistencies in trust calculations due to differences in W_{ns} settings or dynamic adjustment of different nodes, but incurs a large additional data transfer overhead. In this paper, an indirect evaluation approach based on the results of trust calculations is chosen for the actual application scenario characteristics of the terminal self-organizing network.

Assuming that there are two network nodes A and B in a self-organized network, the following steps need to be taken in the process of indirect trust assessment of B by A. A first obtains direct trust assessment data DT from nodes other than itself and B. After getting DT, A needs to rely on the indirect trust correction function $\mathrm{itc}()$ according to the trustworthiness of the data source to correct the DT passed by the node. A simple $\mathrm{itc}()$ can be expressed as

$$\mathrm{itc}(\mathrm{DT}, T) = \frac{\mathrm{DT}_{i\mathrm{B}}T_{\mathrm{B}}}{\sum_{i}^{n-2}T_i}. \tag{2}$$

Also, node A needs to rely on the indirect trust validity function $\mathrm{itv}()$ to evaluate the data validity of the data source

node. For example, a simple itv() based on the distance of the network topology can be expressed in the following form:

$$\mathrm{itv}(\mathrm{DT}, p_i, p_{\mathrm{B}}) = \mathrm{DT}_{i\mathrm{B}}\mathrm{Dis}(i, \mathrm{B}), \tag{3}$$

where the result of $\mathrm{Dis}(i, \mathrm{B})$ is the value corresponding to the distance from i to B in the standardized vector of the inverse correlation dataset of topological distances. Although both itc() and itv() are used to calibrate the acquired DTs, itc() primarily deals with the node's level of trust, i.e., the effect of the node's past behavior on the validity of its DT data, while itv() is used to balance the computational bias of the data affected by the validity of the original data itself used by the source node to compute the direct trust. Together, the calculations of itc() and itv() determine the weight of the different sources of direct trust in the final indirect trust value. In summary, the function of indirect trust assessment of node A on node B in a self-organizing network containing n nodes is described as

$$\mathrm{IT} = \mathrm{ita}(p_{\mathrm{B}}) = \sum_{i=1}^{n-2} \mathrm{itv}(\mathrm{itc}(\mathrm{DT}_{i\mathrm{B}}, T_{\mathrm{B}}), p_i, p_{\mathrm{B}}). \tag{4}$$

The calculation of the indirect trust value reduces the large fluctuations of the trust assessment results within self-organized networks at low data traffic, while resisting some trust spoofing.

Composite trust calculation value of nodes $T_c = W_d\mathrm{DT} + W_i\mathrm{IT}$.

3.2. Trust Data Acquisition. The trust model, as an important basis for trust assessment, establishes uniform standards and methods for assessing trust and determines the mode of operation of the trust assessment. However, status data itself still remains an important factor in determining the validity of trust assessment results. The ability to obtain more valuable data during the operation of a self-organizing network determines the outcome of the trust assessment and the subsequent correctness of route establishment. The value referred to here includes attributes such as timeliness, truthfulness, and objectivity of the data. The present scheme therefore devises a number of methods in the trust data acquisition section to assess the above attributes in addition to methods for obtaining relevant data for trust assessment.

Since this paper focuses on the route establishment problem during data transmission, this scheme chooses wireless transmission as the main way for self-organizing nodes to obtain state data. The wireless data transmission process can be described simply as a node listening for data frames in the wireless medium, retaining the data to be processed and discarding other data frames. The scheme builds on this process by performing further analytical processing of data frames that should have been discarded to obtain basic network state data. The acquisition process can be described as follows. Suppose there are nodes A, B, and C in a self-organized network, and node A and node C need to complete data transmission through node B due to wireless data propagation distance limitation. In this process, the data transmitted by node A needs to pass through the A-B-C transmission path. In this scheme, when A sends a data frame X with destination address C to B, the data sending status $\mathrm{SS} = \{\mathrm{A}_s, \mathrm{A}_d, H, S, T_{\mathrm{send}}\}$ needs to be recorded, where A_s is the source address and A_d is the destination address, H is the data check value, and S is the data frame load size and stores each recorded SS in a data send status list L_{ss}. The state information used for trust assessment is thus obtained through maintenance of the L_{ss} in two main ways: one of them is the timeout record, where A deletes the timeout record and records it as a data transfer timeout event by scanning the L_{ss} at regular intervals and calculating the difference between the current time and T_{send} and comparing it with the preset timeout threshold T'. Its second is forwarding confirmation B, which operates normally during the usual data transfer, and should forward X to C after receiving X. Since the data is transmitted wirelessly, A, which is also in the transmission range of B, will receive an X from B as well. At this point, node A does not directly discard X but records the received information $\mathrm{RS} = \{\mathrm{A}_s, \mathrm{A}_d, H, S, T_{\mathrm{reci}}\}$ about X again. After completing the recording of the RS, A cross-referenced the RS with the SS in L_{ss}. If no corresponding entry is deleted, and if it exists, the difference between T_{reci} and T_{send} is recorded as the transmission delay for this data transfer. A successful data transfer event is recorded and the corresponding entry in L_{ss} is deleted.

For another part of the data, such as the routing information of neighboring nodes and the transmission bandwidth of the data link, limitations due to issues such as network transmission efficiency cannot be obtained through active condition monitoring or operational testing. However, if their data changes are used as some kind of security event triggering mechanism, an incident response mechanism can be reached despite the inability to identify the specific source of the problem. On this premise, it is only necessary to establish a secure information sharing mechanism within the self-organized network and use signature authentication for key information, where the signature authentication only needs to ensure security in a short period of time due to the high timeliness of the state information; in the event of a security incident, information can be shared and cross-referenced to identify the problem and then traced back to the source of the problem based on data signatures.

3.3. Trust Maintenance. The scheme uses a dual event- and time-driven trust update model. In the general mode, the trust value is updated periodically based on the results of statistical analysis of data and historical data within a certain period of time, and at the same time, a number of self-organized network management events will trigger the trust update, in order to respond effectively in the event of relatively specific network security events.

Time-driven periodic trust updates in the scheme are used as a routine maintenance method for the self-organized network trust status, and their main task is to assess the impact of recent network behavior of self-organizing equipment within a specific time frame and in accordance with the design requirements. The main assessment functions include the following: Node Trust Update

Periodic Function uc(), Node Trust Update Function tu(), and parameter selection functions for trust updates. uc() describes the process of calculating the trust update cycle UC and, to some extent, reflects the effective time of the current trust state. The calculation needs to correlate the current overall trust, the raw data, and the previous update interval. The function is described as follows: $\mathrm{UC} = \mathrm{uc}(T_{\mathrm{cur}}, T_{\mathrm{his}}, \mathrm{UC}) = W_T(T_{\mathrm{cur}} - T_{\mathrm{his}})/\mathrm{UC} + \mathrm{UC}_b$. Adjust the size of the trust update interval based on trust trends to balance the overhead caused by trust updates with the corresponding delays. The Node Trust Update Function focuses on the relationship between the current trust value and the historical trust value. Based on the trust update interval, the effective time of the historical trust value can be used as its impact weight on the current trust state when calculating a new trust value to avoid sharp fluctuations in the trust value. Trust assessment results

$$T = \mathrm{tu}(\mathrm{HD}, T_n) = W_{\mathrm{h}} \sum_{i=1}^{\mathrm{len}} \frac{\mathrm{UC}_i T_i}{\mathrm{UC}_t} + W_n T_n. \tag{5}$$

W_h and W_i are the weights of the historical trust value and the current trust value, respectively. The historical trust value HD consists of several sets of historical trust value records and a composite value for the corresponding update interval. Here, the update interval is not the actual update interval time but a calculated value maintained by the trust update module in relation to the actual update interval. This computed value gradually decreases with each trust update event until it falls below a preset threshold and is removed from the computed sequence.

The scheme also includes a trust update mechanism with security event triggers on top of the periodic updates. As mentioned in the periodic update, the scheme designs correlation functions for reducing the additional overhead caused by large fluctuations in trust during system operation, but this also reduces the responsiveness of the model. Due to slower trust changes, it may take longer to dispose of anomalous behavior nodes when they appear in the system. In order to reduce such occurrences, there is a need to establish additional trust renewal mechanisms that are triggered by unforeseen events. The scheme monitors both communication changes and trust threshold events in its implementation.

Communication change monitoring is achieved by monitoring changes in data traffic. As the data production side, the self-organizing gateway will receive relatively stable data uploads from all other nodes in the self-organizing network during the effective operation of the self-organizing network. Therefore, when a gateway node finds that a sensor node has not uploaded data for a long period of time, there is a high probability that the node data transmission will be interrupted or the node will go offline in the network. The program responds in the following ways. First, it will confirm whether the target node is out of the current self-organized network by collecting direct route information for each node and stop responding if the node is normally offline. Otherwise, data transfer tests will be conducted between the target node and the gateway node to find transmission breakpoints and adjust their trust state as well as the data records in the associated node.

Trust threshold events are initiated by each node itself. For a node based on the preset node direct trust thresholds, when the node is found to exist in the operation of a section of the direct trust, the calculation value is lower than the preset value (including the direct trust caused by the communication changes in monitoring the decline in trust) for a number of parameters and data related to trust calculations, including W_d, W_i, and other trust weights and calculated values in HD to be corrected, and triggers a trust update.

4. Route Establishment

4.1. Routing Algorithm. The routing establishment process in this scheme takes the comprehensive trust value output from the trust model as the main parameter, and each sensor node within the self-organized network takes its own direct route as the information basis. The scope of the routing table is gradually extended by exchanging routing information between nodes several times and eventually reaching full reachability with the internal self-organizing network.

Self-organized networks are described in the scheme as a graphical data structure $G = (V, E, W)$. The routing model $G' = (V, E_R, T)$, which is built on top of the network topology, is a directed graph belonging to G. V is the set of points of G', a collection of nodes in a self-organizing network, $p_1 \cdots p_n \in V$. E_R is the edge set of G', which is every single-hop routing path in the routing model $R(a, b) \in E_R$, $a \in V, b \in V, a \neq b$. T is the edge weight of G', which is the path selection weight computed in the routing model based on node trust, path state, and so on. On this basis, a trusted route path can be described by an ordered set of nodes $L_N = [p_1, p_2, \cdots, p_n]$ between the starting node and the target node. The process of establishing such a trusted routing path is based on the calculation and selection of the path selection weights T. The calculation of T has been described in the trust model in Section 3. Each node needs a route score R_s, which is calculated by combining the T, route distance, and other factors of the path nodes, as the basis for route selection. The route scoring function rsc() is as follows: $R_s = \mathrm{rsc}(T, L, D) = w_t T + w_L L + w_d D$. It can be seen that R_s is calculated from the weighted sum of node trust value T, route distance score L, and transmission delay score D. The routing distance and transmission delay are statistical quantities that require a scoring function such as $L = (R_l w + w^2)/R_l$ to unify with the T numerical standard. The calculation weights w_t, w_L, w_d need to be set according to the needs of the route calculation process. Higher values of w_t increase the sensitivity of the route establishment process to changes in self-organized network security conditions and maintain good data transmission quality. A higher value of w_L can reduce the number of forwardings per unit of data in a self-organized network, which helps to reduce the overall data transmission overhead in the network and extend the effective running time. And for w_d, this scheme already includes an evaluation of the transmission delay in the

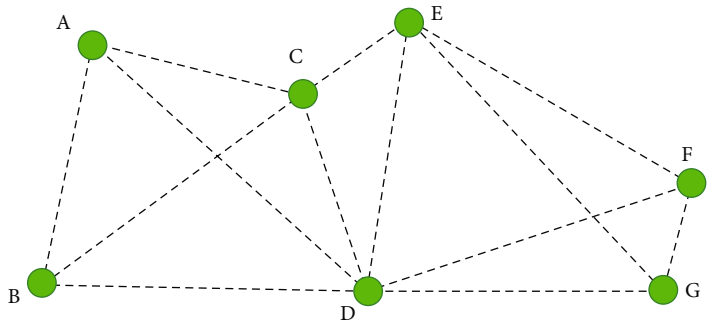


Figure 1: Network topology.

calculation of T. Here, w_d is added only as a run parameter that is closely associated with the routing process, in order to reduce the coupling of the dynamic configuration of the route selection and trust evaluation process and to increase the flexibility of the scheme configuration. w_d can be set to a lower value if there are no special requirements for data transmission delay.

4.2. Route Establishment Process. The route establishment process described in this paper proceeds on the basis of a hypothetical network topology. Suppose there are 7 nodes in the self-organized network, and the network topology is shown in Figure 1. Assume that the self-organized network has been running for some time and that each node in the network has the list of self-organized nodes and the computed trust list. At this point, node A in the network is sending data to node F. The route selection process is described as follows:

(1) A finds out if there is a route entry to F in the list of existing routes and selects the path with the best route score to send if the corresponding route entry exists. If it does not exist, A needs to send a route establishment request rer to its neighbor nodes (i.e., all nodes that can establish a direct route). The rer includes the necessary data such as source node A and destination node F, and the transmission path is shown in Figure 2.

(2) The node receiving the rer responds to the request and looks up the route entry to node F in the local route list. If present, it sends the corresponding route entry to the rer source and ends the response. If not, it sends the rer in the same manner as in (1). Note that nodes responding to rer need to avoid responding repeatedly to the same source node and destination node rer to avoid possible routing loop problems. Figure 3 shows the response of the rer sent by node C.

(3) After the node receives a response from a neighbor node, it needs to perform the following calculations: let A be the sending node of rer and B be the responding node. First, A obtains a comprehensive trust assessment T_B of B from the trust data and then uses the path length of the route entry R of B as the path trust weight and combines the route trust value in R with T_B to obtain the route trust value T for A through B to F. The new route distance and delay are also calculated accordingly. Finally, node A calculates the route score through B to F based on the newly acquired T, L, and D.

(4) When node A receives all of the rer response data, it records the route path with the route score and selects the route entry with the high route score as the path to send data to F to complete the route establishment, as shown in Figure 4. Further screening conditions may need to be established during the selection process to improve the quality of network data transmission. As in this scheme, if the allowable lower limit of the route trust value is set, this route path is directly discarded when the T-value of the existing route entry is below a preset threshold to avoid the security risk associated with a low-trust path.

4.3. Route Maintenance during Data Transmission. Once the route list is established, the node already has the ability to complete the data transfer, but as with the trust evaluation process, the node's route list is also time-sensitive. Therefore, a route maintenance and update mechanism is necessary to maintain the trusted state of the route. Route maintenance in this scheme relies on the triggering of routing-related state changes, specifically trust state changes, route path changes, and relay node state changes.

4.3.1. Trust State Changes. Trust state changes are routing updates that result after a periodic or event-triggered change in a node's trust value. In this case, the node needs to correct the current route list based on the new trust data for the route's trust value and composite score and reselect the preferred route.

4.3.2. Route Path Changes. Route path changes are routing updates that result when a node's direct route changes, where the change can be a direct route change due to a neighbor node going offline or a route change due to path blocking triggered by a trust value below a threshold.

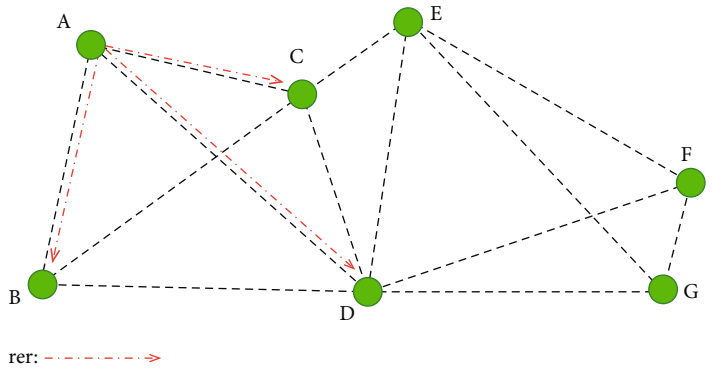


Figure 2: rer transmission path.

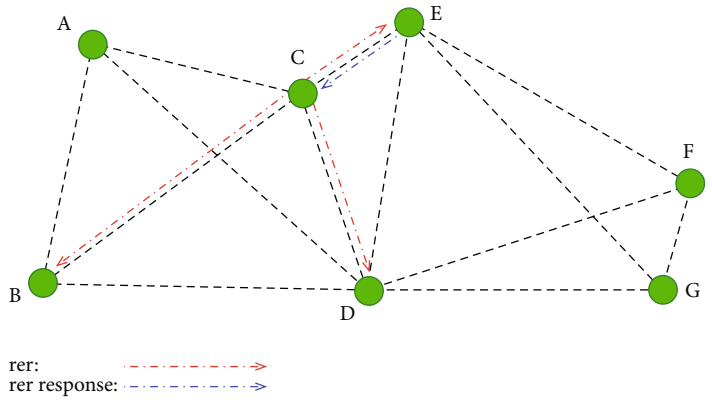


Figure 3: rer response.

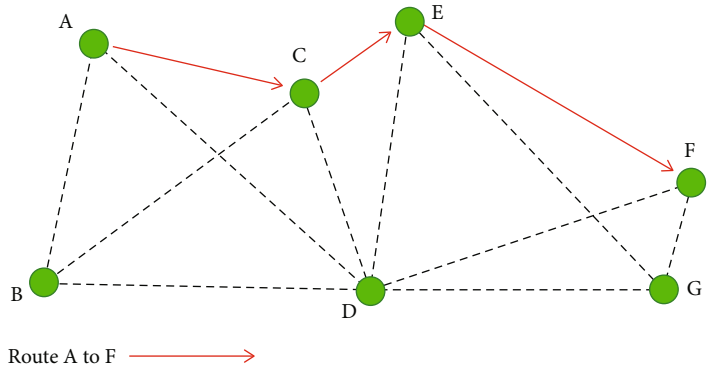


Figure 4: Route path.

4.3.3. Relay Node State Changes. Relay node state changes are the routing updates that result from changes in the routing list of a relay node. After a node within the self-organized network (set this node to A) updates its route list, A broadcasts this route change to its neighbor nodes. If the node B that receives the information meets the following conditions, the routing table contains the route through A and when other alternative routes exist in the routing table. B needs to recalculate the route scores for routes passing through A and update the routing table if a route entry priority change occurs.

5. Performance

In this section, the actual performance of the proposed scheme is tested, including the execution of the model under different parameters and the performance of the model against different forms of network attacks. The experimental environment and data settings other than those described within this section are based on the results of runs of other associated experiments in the laboratory, and a detailed description of these is of little relevance to the main content of this paper and can be considered as initial run parameters

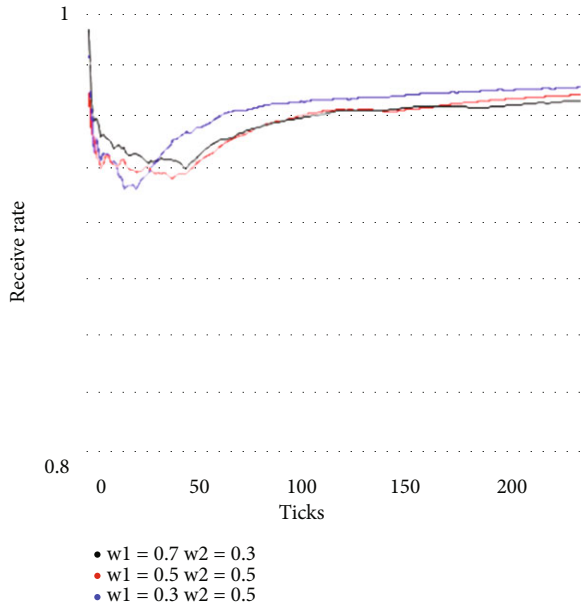


Figure 5: Different W_d and W_i.

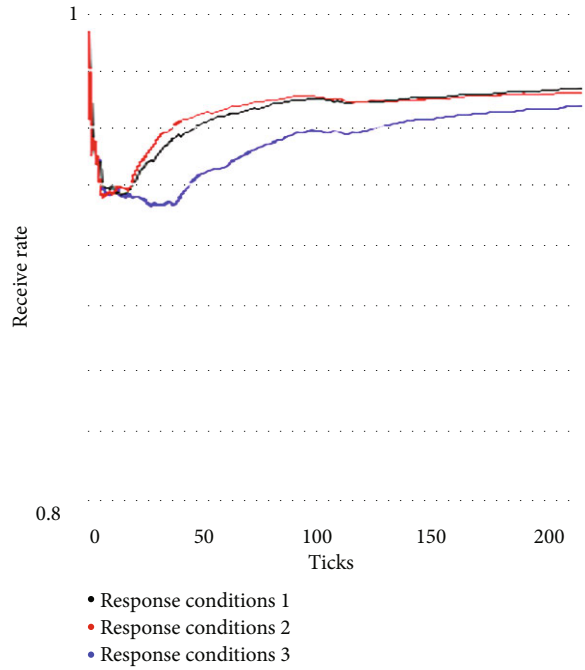


Figure 6: Different response conditions.

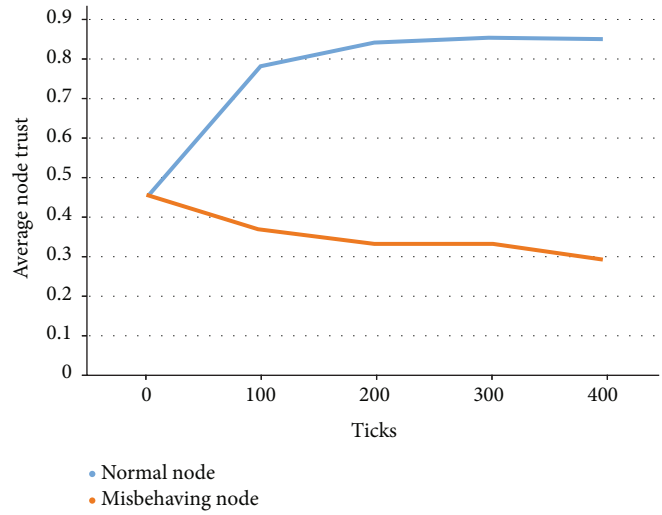


Figure 7: Message alteration attacks.

set randomly within a certain range, with no significant effect on the experimental results.

Figure 5 shows the effect of direct and indirect trust on the change in the overall receive rate of the network during the composite trust calculation process. w1 and w2 correspond to the direct trust value weights W_d and indirect trust value weights W_i, respectively, in the trust model when calculating T_c. Based on the analysis of the changing patterns of the receive rate curves under different parameter configurations, it can be concluded that the higher the direct trust value weights the faster the corresponding rate of composite trust, but the slower trust propagation will lead to a lower rate of its long-term receive rate increase, holding all other parameters constant. The higher the indirect trust value, the faster the trust spreads, but a more severe reduction in receive rate may occur in the short term.

Figure 6 shows the trend of network data transmission rates for different response conditions. The response conditions are of a relatively simple threshold type, where the three curves correspond to the overall data transmission rate from within the group network at threshold values of 0.4, 0.3, and 0.25. By comparison, it can be intuitively understood that the higher the threshold, the more likely the security response is to be triggered, which also makes the node faster to respond to unexpected security events accordingly. The lower the threshold, the more difficult it is to trigger a security response and the slower the node's corresponding response to unexpected security events. However, it should be noted that if we take the curve of Response conditions 2 as a reference, the corresponding curve of Response conditions 3 is not much different from Response conditions 2 in speed compared with Response conditions 1, but the number of emergency responses triggered by Response conditions 3 is much higher than that triggered by Response conditions 2 because it is closer to the initial value of the node trust, which means that Response conditions 3 have paid a lot of computing cost but have no actual effect compared with Response conditions 2. Therefore, it can be understood that the response condition setting needs to be adjusted according to the basic operation of the network, and too high or too low of the trigger conditions will cause the efficiency and stability of the network to decline.

Figure 7 shows the trend of trust changes when the scheme responds to message alteration attacks, where the vertical coordinate is the mean of the composite trust value for the corresponding type of node in the self-organized network and the horizontal coordinate is the time passed since the attack occurred in the self-organized network. From the change of the two lines corresponding to the normal node

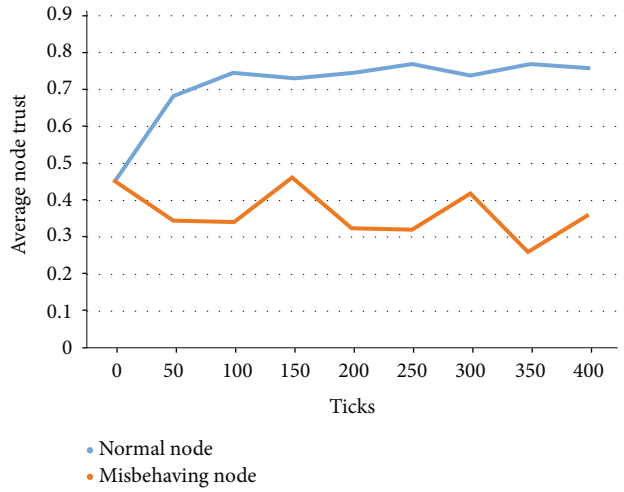


Figure 8: Badmouth attack.

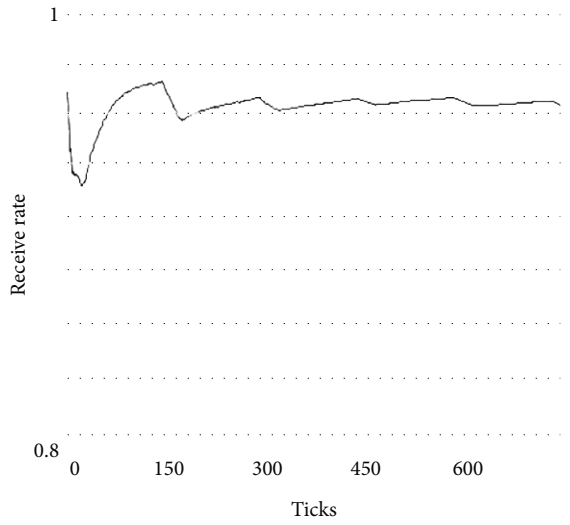


Figure 9: On-off attack.

and the misbehaving node, we can see that after the misbehaving node in a self-organized network attacks and affects the data transmission in the network, the trust evaluation of other nodes in the self-organized network decreases rapidly, which is significantly different from the trust value of other normal nodes.

Figure 8 shows the trend of trust changes in the program's response to the badmouth attack. The model's trust assessment produces large fluctuations when there are misbehaving nodes in the self-organizing network that provide erroneous trust data. However, through the handling of the trust model in the scheme, this trust fluctuation does not affect the distinction between misbehaving nodes and normal nodes when the number of misbehaving nodes in the network is not dominant.

Figure 9 shows the trend in the receipt rate of the program in response to an on-off attack, where the vertical coordinate is the receive rate of the overall data from the self-organized network and the horizontal coordinate is the time since the attack occurred. The misbehaving node periodically changes its own behavior, causing confusion in the node trust assessment process. It can be seen that the network's receive rate drops within a short period of time after an attack is developed but then rises rapidly when other nodes respond accordingly. This makes low-frequency attacks have less impact on the overall operation of the network, while high-frequency attacks are distinguished as misbehaving nodes and isolated, thus limiting the effectiveness of on-off attacks.

6. Conclusion

In sensor self-organized networks, the lack of high-performance base station nodes and the performance limitations of each network node make it impossible to effectively implement information security techniques that rely on encryption and signatures. It is necessary to provide security to the sensor nodes with minimal reliance on encryption and decryption operations. The trust-based trusted routing scheme designed in this paper establishes a more comprehensive and effective trust evaluation system within the self-organizing network through distributed trust evaluation. In terms of energy consumption, the distributed trust evaluation and the passive data collection approach share the consumption of trust computation among the self-organizing network nodes while reducing the additional overhead incurred in the trust evaluation phase. And the mode of adjusting the path selection based on node energy consumption also partially achieves load balancing of data transmission among the self-organizing network nodes. And the effectiveness of the attack form of trust spoofing is reduced by an integrated trust computation with multiple trust sources. The trust and routing distance based routing scheme allows self-organizing networks to efficiently perform the discovery and isolation process on misbehaving nodes completely autonomously, achieving automatic response and effective disposal of multiple forms of network attacks. Nevertheless, the lightweight design of this solution provides limited protection against Blackhole, Sybil, and other forms of attacks on the sensor self-organizing network.

Data Availability

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Project under grant 2019YFB2102303.

References

[1] "Cisco annual internet report (2018–2023) white paper," https://www.cisco.com/c/en/us/solutions/collateral/executive-

perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] R. Mehta, J. Sahni, and K. Khanna, "Internet of things: vision, applications and challenges," *Procedia Computer Science*, vol. 132, pp. 1263–1269, 2018.

[3] A.-S. K. Pathan, Z. M. Fadlullah, S. Choudhury, and M. Guerroumi, "Internet of things for smart living," *Wireless Networks*, 2019.

[4] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem, "Internet of things: evolution and technologies from a security perspective," *Sustainable Cities and Society*, vol. 54, 2020.

[5] The Hacker News, "Dark Nexus: a new emerging IoT botnet malware spotted in the wild," April 2020, https://thehackernews.com/2020/04/darnexus-iot-ddos-botnet.html.

[6] G. J. Chen, *Mobile Internet of Things: Business Model and Case Analysis and Practical Application*, Post & Telecom Press, China, 2016.

[7] J. Cordasco and S. Wetzel, "Cryptographic versus trust-based methods for MANET routing security," *Electronic Notes in Theoretical Computer Science*, vol. 197, no. 2, pp. 131–140, 2008.

[8] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile adhoc networks: a survey," *IEEE Communication Surveys and Tutorials*, vol. 14, no. 2, pp. 279–298, 2012.

[9] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: attack analysis and countermeasures," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 867–880, 2012.

[10] J. Guo, I.-R. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in Internet of things systems," *Computer Communications*, vol. 97, pp. 1–14, 2017.

[11] J.-H. Cho, A. Swami, and I.-R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 562–583, 2011.

[12] F. Li and J. Wu, "Uncertainty modeling and reduction in MANETs," *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 1035–1048, 2010.

[13] W. Li, A. Joshi, and T. Finin, "Coping with node misbehaviors in ad hoc networks: a multi-dimensional trust management approach," in *2010 Eleventh International Conference on Mobile Data Management*, pp. 85–94, Kansas City, USA, 2010.

[14] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pp. 1238–1246, Phoenix, AZ, USA, 2008.

[15] P. B. Velloso, R. P. Laufer, D. de O. Cunha, O. C. M. B. Duarte, and G. Pujolle, "Trust management in mobile ad hoc networks using a scalable maturity-based model," *IEEE Transactions on Network and Service Management*, vol. 7, no. 3, pp. 172–185, 2010.

[16] S. Adali, R. Escriva, M. K. Goldberg et al., "Measuring behavioral trust in social networks," in *2010 IEEE International Conference on Intelligence and Security Informatics*, pp. 150–152, Vancouver, BC, Canada, 2010.

[17] T. DuBois, J. Golbeck, and A. Srinivasan, "Predicting trust and distrust in social networks," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pp. 418–424, Boston, MA, USA, 2011.

[18] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Computing Survey*, vol. 45, no. 4, 2013.

[19] S. Yi, P. Naldurg, and R. Kravets, "A security-aware routing protocol for wireless ad hoc networks," in *Stochastic Analysis in Discrete and Continuous Settings*, Springer, 2002.

[20] A. Tajeddine, A. Kayssi, A. Chehab, I. Elhajj, and W. Itani, "CENTERA: a centralized trust-based efficient routing protocol with authentication for wireless sensor networks," *Sensors*, vol. 15, no. 2, pp. 3299–3333, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6668478, 15 pages
https://doi.org/10.1155/2021/6668478

Research Article

Just Shake Them Together: Imitation-Resistant Secure Pairing of Smart Devices via Shaking

Congcong Shi,[1,2] **Lei Xie**,[1] **Chuyu Wang**,[1] **Peicheng Yang**,[1] **Yubo Song**,[3] **and Sanglu Lu**[1]

[1]*State Key Laboratory for Novel Software Technology, Nanjing University, China*
[2]*State Grid Key Laboratory of Information and Network Security, Global Energy Interconnection Research Institute Co. Ltd, China*
[3]*School of Cyber Science and Engineering, Southeast University, China*

Correspondence should be addressed to Lei Xie; lxie@nju.edu.cn

Received 10 December 2020; Revised 2 February 2021; Accepted 23 March 2021; Published 5 April 2021

Academic Editor: Zhipeng Cai

In traditional device-to-device (D2D) communication based on wireless channel, identity authentication and spontaneous secure connections between smart devices are essential requirements. In this paper, we propose an imitation-resistant secure pairing framework including authentication and key generation for smart devices, by shaking these devices together. Based on the data collected by multiple sensors of smart devices, these devices can authenticate each other and generate a unique and consistent symmetric key only when they are shaken together. We have conducted comprehensive experimental study on shaking various devices. Based on this study, we have listed several novel observations and extracted important clues for key generation. We propose a series of innovative technologies to generate highly unique and completely randomized symmetric keys among these devices, and the generation process is robust to noise and protects privacy. Our experimental results show that our system can accurately and efficiently generate keys and authenticate each other.

1. Introduction

More and more smart mobile devices have been used in our daily life. These devices usually communicate with each other via Bluetooth or WiFi Direct. However, data transmitted through these methods is not completely private since the attacker can intercept information by eavesdropping in the wireless channel. This leads to insecure transmission of private data such as an electronic business card. Therefore, it is essential for users to authenticate each other and establish spontaneous secure connections between devices. This is to achieve mutual authentication and key generation.

Existing solutions mainly utilize the password or patterns to authenticate each other and establish secure connections. However, it is not suitable to apply these solutions on mobile devices. These solutions have three major weaknesses. First, for some small devices like smart bracelets, they have no touch screens for users to enter the password or patterns. Second, the password/patterns are vulnerable to shoulder surfing attacks because they are easily peeped in public places. Third, users prefer to choose shorter passwords or simpler patterns if they have to input passwords or patterns frequently; this leads to a higher risk of information leakage.

Recently, some emerging work tries to make use of the embedded sensors such as touch screens [1–5], accelerometer [6–15], gyroscope [16–19], and wireless signals [20–26] to capture user's behavior and realize mutual authentication and key generation. Specifically, the accelerometer is usually used to authenticate devices and generate keys by shaking. In such scenarios, the user first needs to put together the devices to be authenticated and then shake them simultaneously. The accelerometer can collect the movement features of these devices during shaking process. The detailed profile and movement pattern of each device can be extracted from the sensor data. The devices can detect if they are shaken together by comparing these information. These

devices will authenticate each other and generate the unique and consistent key only when they are shaken together. The basic idea is that users tend to produce very random but similar trajectories for multiple devices during shaking process. This behavior contains enough features to create a unique profile for these devices, and it is difficult for attackers to mimic this behavior.

However, these existing solutions have the following limitations: first, the key generation is inefficient since sensor data is temporally and spatially correlated. The user has to shake devices long enough to generate enough bits. Second, they envision the ideal application scenario, where the devices are all homogeneous. However, the devices we use to pair are not always the same. These devices may differ in size, measurement units, and sampling frequency. Third, the sensors they use can only make coarse-grained perceptions of user's behavior. This leads to weak defenses against shoulder surfing attacks since it is easy to be imitated.

In this paper, we propose iShake, an imitation-resistant secure pairing framework including authentication and key generation for smart devices, by shaking these devices together as shown in Figure 1. iShake solves the limitations of previous work. First, we eliminate spatiotemporal correlation by selective sampling of sensor data. Moreover, we utilize hash-based message authentication code (HMAC) to generate random bits. Second, we investigate the key generation in heterogeneous devices. We find the "device displacement" which is common in shaking heterogeneous devices. We effectively solve this problem through adaptive calibration and quantization. Third, we capture the user's behavior in a much finer granularity by utilizing both the accelerometer and gyroscope. Therefore, we can perceive the micromovement of user's behavior. We adaptively sample sensor data from different dimensions to generate more differentiated bits according to the different sensitivities of different dimensions.

2. Related Work

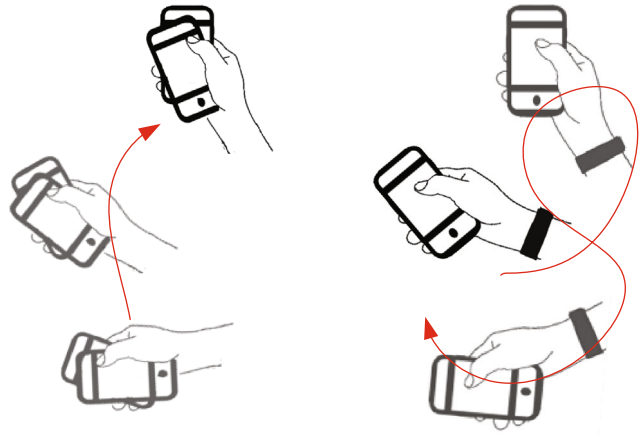
Nowadays, many researchers focus on how to effectively conduct pairing or authentication in smart devices, by using equipped sensors like touch screens [1–5], accelerometer [6–14], gyroscope [16–19], and wireless signals [20–26]. The idea of associating two or more devices by shaking them together was first put forward in "Smart-Its Friends" [27]. It uses shared "context proximity" of dedicated devices to authenticate each other, which is created by explicit user-controlled action. They mainly leverage the collected sensor data to carry out pairing or simple authentication in smart devices, without further considering establishing secure connections. Besides, they all try to exchange the raw sensor data or explicit feature information to each other, which cannot guarantee confidentiality and lead to heavy transmission overhead.

Recent work considers to further establish spontaneous secure connections among the smart devices. Mayrhofer proposes a protocol for generating secret shared keys from similar sensor data streams [7]. Rene and Gellersen leverage traditional asymmetric encryption and the feature of raw sensor data to authenticate dedicated devices and create shared keys for secure channel between shaking devices [8, 9]. However, these methods mostly leverage asymmetric encryptions to encrypt raw sensor data, which brings heavy computational cost and cannot be effectively implemented in current mobile devices like the smart bracelets. Xu et al. [28] propose a novel secret key generation protocol, GaitKey, which can generate the same cryptographic key based on the unique gait pattern for two legitimate devices of the user. Moreover, Revadigar et al. [29] further leverage the accelerators of multiple devices attached on the human body and generate the common secret keys based on the unique walking style of the user. But these works focus on the gait pattern for key generation between multiple devices of the same user, instead of the devices of different users. The most closely related works are [30, 31]. In [30], Bichler et al. present a novel approach to establish a secure connection between two devices by shaking them together. Instead of distributing or exchanging a key, the devices independently generate a key from the measured acceleration data by appropriate signal processing methods. In [31], Bichler et al. further propose an algorithm to independently synchronize the shaken devices so that reliable key generation becomes possible. However, their method requires an additional training phase to extract the detailed profile and pattern of user behaviors on the specified devices, which severely limits its application for different devices and different users. Moreover, some recent works try to build a secure connection between two devices (e.g., smartwrist or smartwatch) based on the handshaking gestures [13, 32]. In this paper, we focus on shaking two smartphones together to generate the cryptographic keys, which can generate more stable and distinguish keys compared with the handshaking.

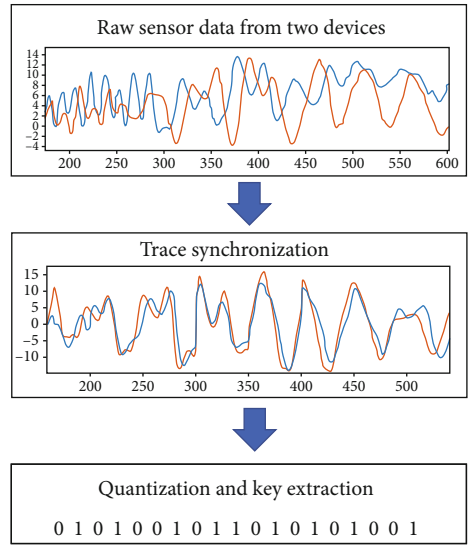
3. Understand Shaking via Empirical Study

To understand how to conduct efficient mutual authentication and key generation, while considering the shaking gestures, shaking frequency, and types of sensors, we conduct comprehensive experiments in real environment, illustrate several novel observations, and extract some important clues. In the following experiments, we sample the sensor data on Samsung Galaxy Nexus i9250 and use the accelerometer and gyroscope to continuously collect the sensor data during the shaking process. Both the accelerometer and gyroscope can collect the trace of user behavior in three dimensions (x-, y-, and z-axes). For brevity's sake, we, respectively, use A_x, A_y, A_z and G_x, G_y, G_z to denote the accelerometer data and gyroscope data in various dimensions. The sensor data from accelerometer and gyroscope are, respectively, measured in the units of meter per second squared and radians per second. We totally record 100 sets of shaking traces at 100 Hz, and each set consists of 2048 data points.

Without loss of generality, when sampling sensor data, we shake devices using the following patterns of shaking gestures: (1) *back and forth*: the user shakes the devices in a back-and-forth approach with a large movement range of arm; the linear acceleration should be fairly large; (2) *roll*: the user shakes the devices with some micromovement of



(a) Shake two phones together (b) Shake a phone and a smart bracelet together

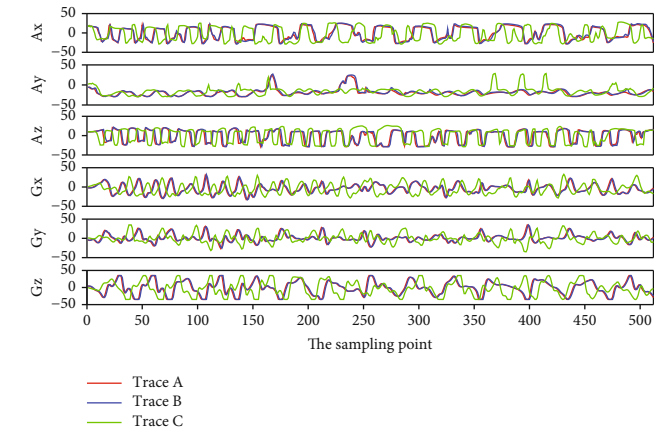


(c) Key generation

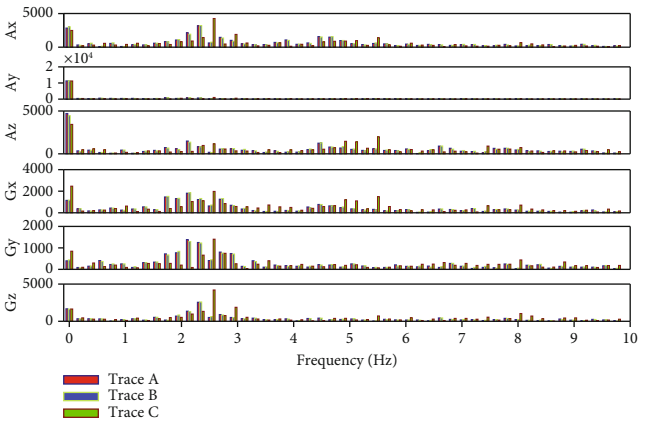
Figure 1: Application scenario of iShake.

the wrist; the angular acceleration should be fairly large; (3) *cocktail*: it is a combination of back and forth and roll, so both the linear acceleration and the angular acceleration should be fairly large.

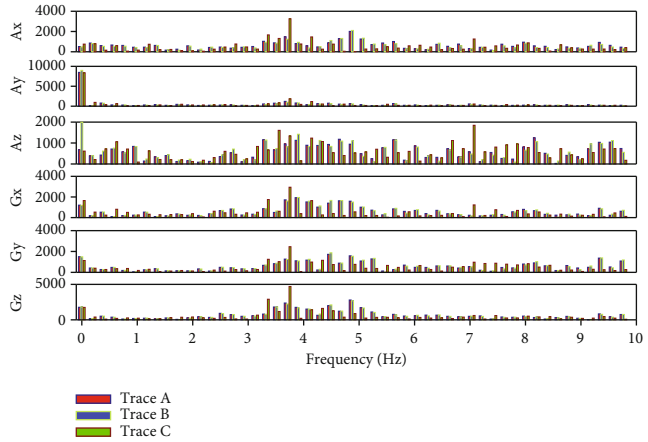
3.1. Sensor Data in Time Domain and Frequency Domain. The sensor data sampled when shaking together have very close patterns, while sensor data sampled when shaking independently always have difference in details. Specifically, the sensor data in the time domain can effectively verify the difference of user behavior during shaking, while the sensor data in the frequency domain can reveal some statistic regularities like the activity degrees of user behavior. We obtain the sensor data by trying three different patterns of shaking gestures (back and forth, roll, and cocktail). We, respectively, test the case of shaking together (traces A and B) and shaking independently (traces A and C) and try to make the shaking gesture as random as possible. We find that for all shaking patterns, the sensor data sampled when shaking together have very close patterns, while sensor data sampled when shaking independently always have difference in details. This just provides us a chance to effectively differentiate multiple devices that are not shaking together. Figure 2(a) shows an example of detailed results from the accelerometer and gyroscope in the x-, y-, and z-axes in the time domain for cocktail. Figures 2(b)–2(d) further show the sensor data in the frequency domain, which is obtained from fast Fourier transformation (FFT) over the raw sensor data in the time domain. The sensor data in the frequency domain show some regularities in a statistical approach. Specifically, if the sensor data contain more components in higher frequency, it means that the user has more detailed activities during the shaking process. We find that the shaking frequency mainly lies in between 0 Hz and 10 Hz, and the sensor data generated from the roll and cocktail pattern have more signals in higher frequency than back and forth; this implies that these shaking patterns can generate more details during the shaking process, which would help generate more number of bits for the key generation. Obviously, cocktail is the most efficient pattern in generating randomized bits, since it has the most details in degree of activity.



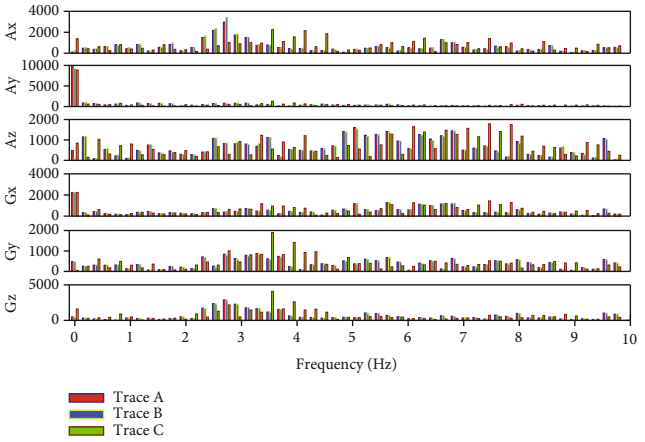
(a) Sensor data in the time domain (cocktail)



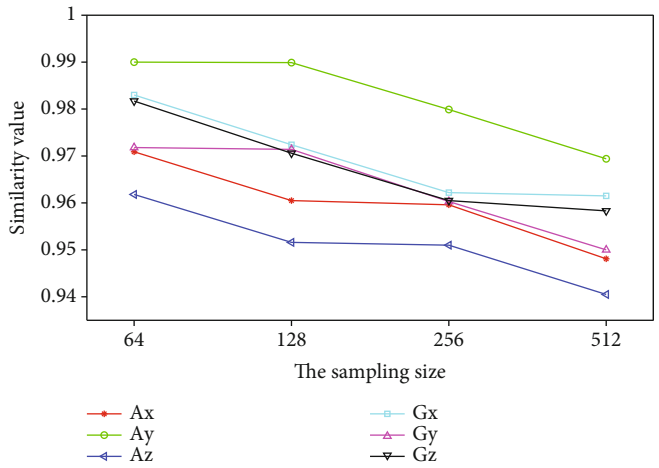
(b) Sensor data in the frequency domain (back and forth)



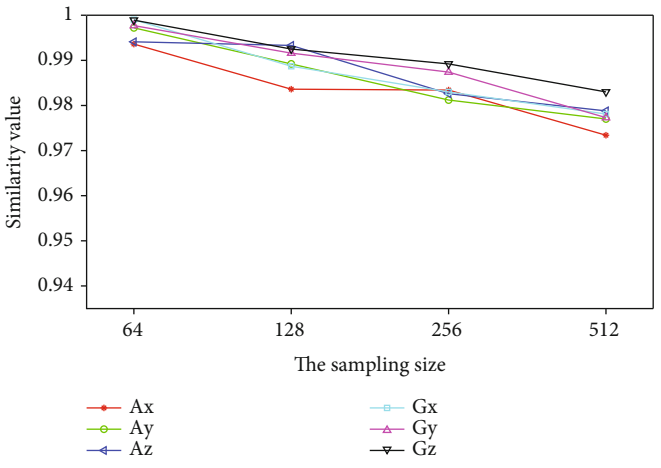
(c) Sensor data in the frequency domain (roll)



(d) Sensor data in the frequency domain (cocktail)

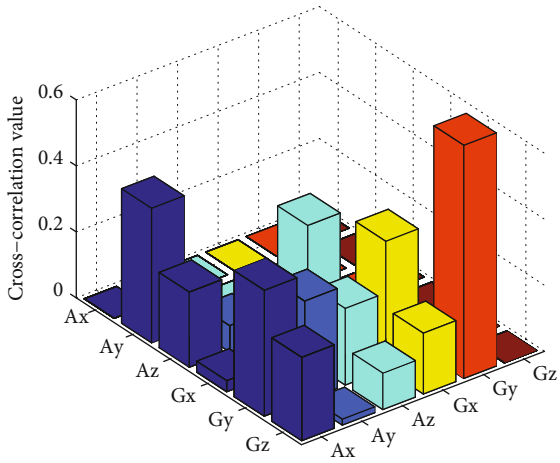


(e) Similarity in the time domain (cocktail)

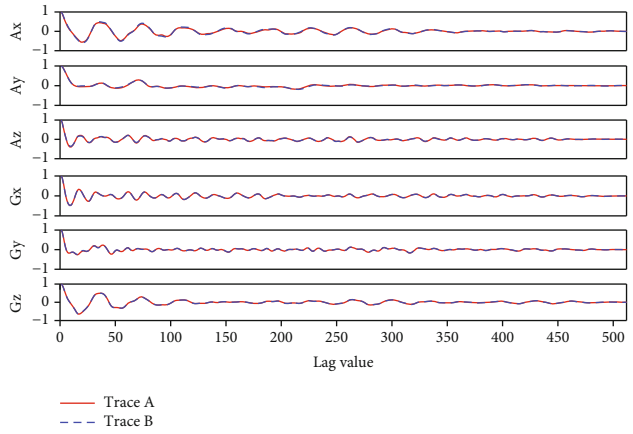


(f) Similarity in the frequency domain (cocktail)

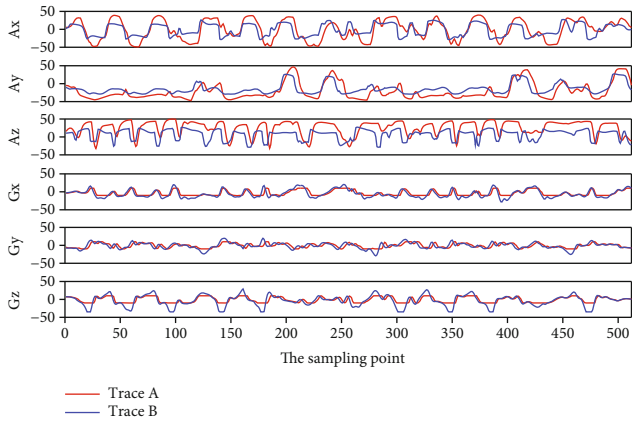
Figure 2: Continued.



(g) Correlation in the space domain (cocktail)



(h) Correlation in the time domain (cocktail)



(i) Sensor data in the time domain (cocktail)

Figure 2: Analysis on the sensor data in various domains.

3.2. Similarity in Time Domain and Frequency Domain. When we use a window to sample the raw sensor data from devices shaking together, we find that as the size of window increases, the similarity of the traces gradually decreases. It is essential to set an appropriate granularity for the window to avoid too many mismatched bits. For two traces generated by shaking together, we, respectively, divide the series of sensor data into a number of blocks with fixed size in regard to each dimension. Figures 2(e) and 2(f), respectively, show the similarity in all dimensions for various block sizes in regard to the time domain and frequency domain. As we increase the block size from 64 bits to 512 bits, we find that the average similarity in all dimensions gradually decreases. The reason is as follows: as the amount of raw sensor data increases with the block size, the amount of mismatched data between the two pairs increases; hence, the similarity of the two pairs of sensor data sequence decreases. This implies that, if we need to verify the consistency of the generated bits in key generation, we need to set an appropriate granularity for the window to avoid too many mismatched bits. Note that the similarity in the time domain is always lower than the frequency domain; this means that the sensor data in the time domain can more effectively help verify the difference of user behavior than frequency domain.

3.3. Sensitivity in Various Sensors and Dimensions. Sensors in different dimensions have different sensitivities to capture the user behavior during the shaking process; the gyroscope is more sensitive than the accelerometer in all three dimensions for depicting the user behavior. According to the results in Figures 2(b)–2(d), it is noted that the sensors have different values for different dimensions in the frequency domain; this means that the users have different degrees of activities in different dimensions. For example, in Figure 2(d), the accelerometer data in the y-axis is apparently smaller than the accelerometer data in the x -axis and z-axis. This is understandable since any kind of shaking pattern may have intense activities in some dimensions and weak activities in the other dimensions. Therefore, the sensor data in different dimensions have different sensitivities to capture the user behavior during the shaking process, and the sensor data in the frequency domain can rightly reveal these different sensitivities. Moreover, we find that the gyroscope is usually more sensitive than the accelerometer in all three dimensions; this happens in all three kinds of shaking patterns. We believe this is because the micromovement of the wrist is usually larger than the movement of the arm during the shaking process.

3.4. The Correlation in Space and Time Domain. The sensor data have correlations in both the space and time domains, especially when the devices are shaking in a regular approach. In order to study the correlation in the space and time domain, we shake the devices with the cocktail pattern as randomly as possible. In Figures 2(g) and 2(h), we, respectively, conduct correlation analysis of sensor data in both the space and time domains. We use the Pearson product-moment correlation coefficient to calculate the correlation: $\rho_{x,y} = (E[(x-\mu_x)(y-\mu_y)])/\sigma_x\sigma_y$, where μ_x, μ_y are, respectively, the mean of x and y; σ_x, σ_y are, respectively, the standard deviation of x and y; and E is the expectation.

Figure 2(g) shows the correlation in the space domain; we compare the correlations of sensor data from the accelerometer and gyroscope in all three dimensions. We find that there indeed exist some correlations among these sensor data. For example, the correlation is close to a value of 0.4 for several dimensions in both accelerometer and gyroscope. The reason is that, when the user is shaking, the sensors like the accelerometer and gyroscope may capture the same activity simultaneously from all dimensions. Figure 2(h) shows the correlation in the time domain; we conduct crosscorrelation in the same trace for different dimensions. We find that, even when we shake the devices in a rather random approach, there still exist some correlations for the traces in the time domain. Therefore, in order to ensure the randomicity in key generation, it is essential to remove the redundancy in both the space and time domains. Moreover, we find that as the sensor data are having more components in higher frequency, the value of autocorrelation is getting smaller. The reason is that the proportion of components in high frequency is positively relevant to the complexity of the raw sensor data; when the raw sensor data is becoming more complex, the value of autocorrelation is getting smaller. Therefore, we can sufficiently leverage the sensor data with more details in high frequency to improve the randomness.

3.5. Shaking with Heterogeneous Devices. While shaking heterogeneous devices, it is common to see the device displacement; i.e., the devices may have relative displacement during the shaking process due to the variation in sizes; this may cause these devices to be out of sync and have biases in various dimensions. In order to study the moving patterns while shaking with heterogeneous devices, we shake two heterogeneous devices (device A: Samsung Galaxy Nexus i9250, 4.65 inches; device B: MI 2S, 4.3 inches) together with the cocktail pattern. Figure 2(i) shows the raw sensor data in the time domain. Different from the previous observations, we find that the traces may have some clear differences in amplitude with respect to some dimensions, especially for the sensor data from the accelerometer. The traces in the gyroscope only have some slight difference in amplitude as compared to the accelerometer. The reason is as follows: while shaking heterogeneous devices, it is common to see that the devices may have relative displacement due to the variation in sizes. Hence, there might be some differences in regard to the sensor data in all dimensions. As the displacement issue mainly impacts the parameters in linear acceleration, so it mainly leads to disturbance in the sensor data from the accelerometer. Nevertheless, as these devices are shaking in the same trajectory, the general outline of the moving trace remains unchanged; there still exists much similarity with respect to the sensor data in variation trend.

4. iShake Framework

4.1. Design Goal and Challenges. The key goal is to conduct mutual authentication and generate symmetric key over insecure channel via shaking smart devices. Each device can independently generate a common key by exchanging a limited number of feature data instead of the whole series of raw sensor data, so that each device can establish a privacy-preserving channel with this common key between each other. According to the experimental study, we find that there are many disturbing factors that hinder devices to generate a unique and consistent key. We still need to effectively tackle the following challenging problems: (1) *generate distinctive bits*: in order to be imitation-resistant, the device should generate more distinctive bits to differentiate the legal users and the attacker. The generated key should be highly distinctive by exploring the characteristic details from the user's moving gesture. (2) *Robust to the noise*: due to the natural deviations and the device displacement in real applications, we cannot guarantee that any two traces have identical trajectories, even if the devices are shaking together. There are always some differences in details of the traces. The common features should be effectively extracted to generate a consistent key. (3) *Reduce the correlations*: there exist some correlations for the raw sensor data in both the time and space domains; the generated key should be made fully randomized to increase entropy by sufficiently reducing the relevant correlations and redundancies.

4.2. Threat Model. We assume attackers cannot hack into the target mobile devices to obtain the raw sensor data for user behaviors during the shaking process. Nevertheless, we assume attackers have the following three capabilities. First, attackers can launch eavesdropping attack by intercepting the transmitted messages in the open environment. Second, attackers can launch man-in-the-middle attack between two devices in interaction. Third, attackers can conduct shoulder surfing by privately imitating the legal user's behaviors in handshaking.

4.3. Mutual Authentication and Key Generation Framework. Figure 3 shows the diagram of the mutual authentication and key generation framework. The framework mainly includes the following steps: *data calibration*, *quantization*, and *key extraction*. The user first shakes the devices together in an arbitrary approach. Once detecting the event of shaking, each device starts to sample the user's behavior via the accelerometer and gyroscope. In the step of *data calibration*, the devices adaptively conduct *trace synchronization* and *data interpolation* on the sampled raw sensor data. In this way, the sensor data can be effectively synchronized and smoothed to mitigate the impact of out-of-sync and device displacement. In the step of *quantization*, each device independently generates

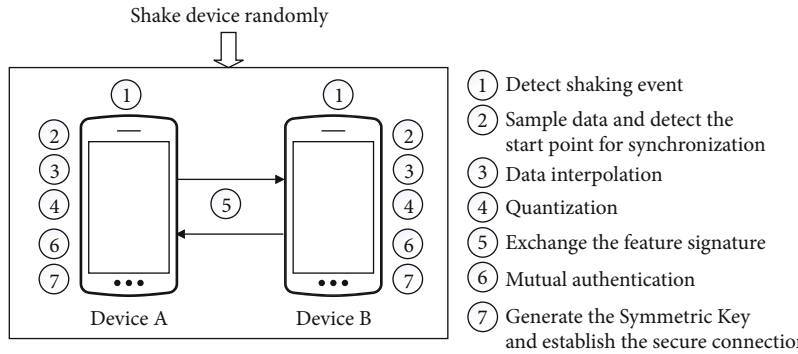


Figure 3: The diagram of mutual authentication and key generation.

bit series from the calibrated sensor data, according to the sensitivity of user behavior in different dimensions. In the step of *key extraction*, the devices exchange a limited number of encrypted messages among each other, so as to verify the consistency for mutual authentication and selectively use the consistent bits for key generation. In this way, these devices can generate a unique and consistent key via shaking.

4.4. Data Calibration

4.4.1. Trace Synchronization. In order to generate a consistent key among the shaking devices, it is essential to first synchronize the locally generated raw sensor data traces among each other, so as to make it possible to generate identical bit series for candidate key usage.

Conventionally, the synchronization is realized by broadcasting a *beacon* message from a specified device to the other devices. However, due to the existence of the attacker, it is impossible for a device to differentiate between the legal device and an illegal device, since the legal devices do not share a secret key in advance. An illegal device can thus send multiple *beacon* messages to interfere the synchronization among those legal devices. Therefore, we can only rely on the inherent *beacons* inside the raw sensor data traces to devise efficient synchronization method for shaking devices.

Figure 4 shows an example of trace synchronization. Note that before the user shakes the devices, the sensor data in all dimensions remains very close to 0. Once the user starts to shake the devices, the sensor data in all dimensions start to increase almost simultaneously. This provides us an opportunity to use it as an *inherent beacon* to locate an identical start point for synchronization. However, since we learn from the experimental study that the sensor data in different dimensions have different sensitivities to the user behavior, we need to comprehensively investigate the sensor data in all dimensions. In Algorithm 1, we propose an efficient solution for trace synchronization. In this algorithm, for each dimension, we, respectively, use a sliding window to simultaneously evaluate the sensor data; once we detect that the average value inside the window is beyond a certain threshold, we record the start point of the window. Realizing the different sensitivities in different dimensions, we use the average value of the first k points as the start point of the sensor data sequence. In this way, the traces from different devices can be effectively synchronized.

In order to conduct accurate trace synchronization, it is important to set an appropriate window size for the sliding window. If the window size is too large, then the sliding window cannot be very sensitive to the sudden increase of the sensor data; if the window size is too small, then the sliding window cannot tolerate any unpredictable turbulence or noise from the outside. According to the experimental study, we note that the user's shaking frequency is usually below 10 Hz; we use f_{sk} to denote the average frequency, e.g, $f_{sk} = 5$ Hz. We then use f_{sk} to denote the sampling frequency, e.g., $f_{sp} = 100$ Hz; thus, an appropriate value for window size should be $w = f_{sp}/f_{sk} = 20$. This implies that the window can effectively capture a complete action for average cases.

4.4.2. Data Interpolation. After the *trace synchronization*, the generated raw sensor data can be effectively synchronized. However, according to the experimental study, it is noted that due to the natural deviations and device displacement, any two traces are not necessary to have identical trajectories, even if the devices are shaking together. There are always some differences in details of the traces. Therefore, in order to make it robust to the noises, it is essential to conduct *data interpolation* to retain the broad outline while dropping away those details.

A key challenging problem is how to effectively differentiate between the outline and the detail. Since the user may shake the devices in either a rapid or slow approach, the major components of the raw sensor data may either in high frequency or in low frequency. Hence, we need to conduct data interpolation by efficiently differentiating between the outline and the detail based on the user's actual behavior.

As the user's shaking behavior has some locality in the time domain, e.g., the user's shaking frequency may not change too much within a limited time interval, we first break the whole data sequence S into multiple subsequences D; then, we apply data interpolation to each subsequence D. For each subsequence D, we maintain a sliding window with a window size w to smooth the sensor data. The window size w can be adaptively adjusted to fit the actual situation of the trace. The process of data interpolation is shown in Algorithm 2. While using the sliding window to smooth the sensor data, we adaptively adjust the window size w by iteratively evaluating the crosscorrelation coefficient θ against a certain threshold t, e.g., 90%. Once the

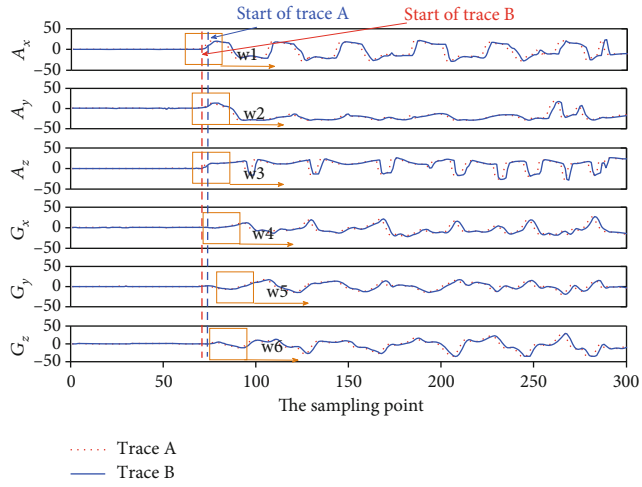


Figure 4: Detect an identical start point for synchronization.

crosscorrelation coefficient is approaching the threshold, the smoothing process is terminated.

Figure 5 shows an example of the data interpolation. For two traces A and B obtained by shaking together, it can be found that before interpolation, the two traces have a number of differences in detail, which hinders them from generating identical bit series. We thus set the threshold $t = 0.9$ and adaptively adjust the window according to the trace; finally, it converges to a window size of 16. We find that, after the interpolation, the different details can be effectively smoothed and the major outline can be better retained as compared to the case with a window size of 32.

4.5. Data Filtering. Data filtering is a technique for improving the quality of motion sensor data. This technique selects the appropriate frequency band for fine-grained data filtering to reduce noise in the motion sensor data. This method can be applied in the training phase to determine the most efficient filtering method for noise reduction.

The proposed data filtering technique selects bands of different lengths rather than a fixed low-pass or high-pass band. This technique divides the full frequency band into several subbands whose granularity is determined by cutting the entire frequency band and reconnecting according to certain rules. These subband samples are used to filter the raw sensor data and to screen out the best performing subbands.

The most commonly used band division method is the $1/n$ Octave method, which divides the entire frequency band into two halves, then divides the low-frequency half into two halves and iterates according to this law. Finally, the method divides each subband equally into n new subbands. The $1/n$ Octave method is characterized by dividing the low-frequency band into finer-grained subbands that facilitate the processing of low-frequency dominant raw data. The 1/2 Octave method is used to divide the entire frequency band into 16 subbands. Since the first three low-frequency subbands are smaller in size, they were combined into one subband.

After obtaining the 14 subbands divided by the Octave method, the 14 subbands are spliced and cascaded into different granularities. Finally, a total of 105 frequency band samples of different lengths are obtained by this method. Then, these samples are applied to an infinite impulse response filter and evaluate the performance of these subbands to filter the motion sensor data.

4.6. Quantization. After the data calibration, we need to quantize the calibrated sensor data sequence into bit series for key generation. In order to boost the bit generation rate, we consider to quantize the amplitude of the sensor data across different sensors and dimensions. According to the experimental study, it is noted that different sensors have different sensitivities to the user's behavior in different dimensions. Therefore, in order to generate more distinctive bits based on user's profile, it is essential to generate bits from those dimensions which is more sensitive than the others.

Then, it raises a new challenge for designing the quantization algorithm: how to effectively evaluate the sensitivity to user's behavior for various dimensions? As a matter of fact, according to the experimental study, the sensor data in the frequency domain can reveal the activity degree of user behavior in different dimensions. Therefore, we can conduct *Fourier analysis* over the calibrated data to obtain the activity distribution in frequency.

Algorithm 3 shows the details of the quantization. When we get the calibrated sensor data from different dimensions, we first conduct fast Fourier transformation (FFT) over them for each dimension i. Then, we compute the top-k principal components in the frequency domain and get the highest frequency of these components as f_i. This implies that the user's behavior is most active within the band $[0, f_i]$. According to the Shannon theorem, we use the sampling frequency $2f_i$ to quantize the data sequence. In this way, the sensor data in those sensitive dimensions can effectively assist in generating more distinctive bits based on the user's behavior. We then normalize the sampled data such that the gap between two traces by shaking together can be effectively reduced; we then apply the level crossing method to generate the bit series according to the specified quantization levels.

After that, for each dimension i, we, respectively, break the generated bit series S_i into blocks with variable size according to the specified time line; the exact size right depends on the number of generated bits for each dimension. We then merge these blocks from multiple dimensions to bit series S in time order. In this way, we can generate a candidate bit series for the following key extraction. Figure 6 shows an example for quantization from different dimensions based on different sensitivities.

4.7. Mutual Authentication and Key Extraction. After the step of quantization, we can obtain a bit series according to the calibrated sensor data. However, these bit series cannot be directly used as the final key, because of the following reasons: (1) nonrandomness: due to the correlations in the space and time domain, there exist redundancy and correlation for a substantial part of the bit series. This causes that the generated key does not have good statistical randomness and high entropy. (2) Inconsistency: due to the issues like device displacement and noises, there still exist mismatched bits in

Require: 1) The raw sensor data sequence: D. 2) The threshold for accelerometer and gyroscope: α_a and α_g.
Ensure: The start point of the sensor data sequence: P.
1: Initialize a sliding window W_i for each dimension i, set the window size to $w = f_{sp}/f_{sk}$. Initialize the threshold t_i for each dimension i according to the sensor type.
2: **for** each dimension i**do**
3: Forward the sliding window W_i along the raw sensor data sequence D, calculate the average value inside the window W_i as v_i.
4: **if** $v_i > t_i$ **then**
5: Set the start point of the window as P_i. Stop forwarding the window W_i.
6: Find the first k points among the point set $\{P_i\}$ for all dimensions. Set the average value of the k points as P.

Algorithm 1: Trace synchronization.

Require: 1) The synchronized raw sensor data sequence: D. 2) The threshold of the cross correlation coefficient: t.
Ensure: The interpolated sensor data sequence: D_f.
1: Set the window size $w = 1$. Set the index of the sliding window $i = 1$. Set the cross correlation coefficient $\theta = 1$.
2: **while** $\theta > t$ **do**
3: Set the window size $w = w \cdot 2$.
4: **for** $i \leq$ the length of D **do**
5: Smooth the value D_i to $D_{f,i}$ with the average value $\bar{D}$ in the sliding window.
6: Forward the sliding window, set $i = i + 1$.
7: Conduct cross correlation between D and D_f, get the coefficient θ.
8: Set the window size $w = w/2$. Smooth the sequence D to D_f with the sliding window.

Algorithm 2: Data interpolation.

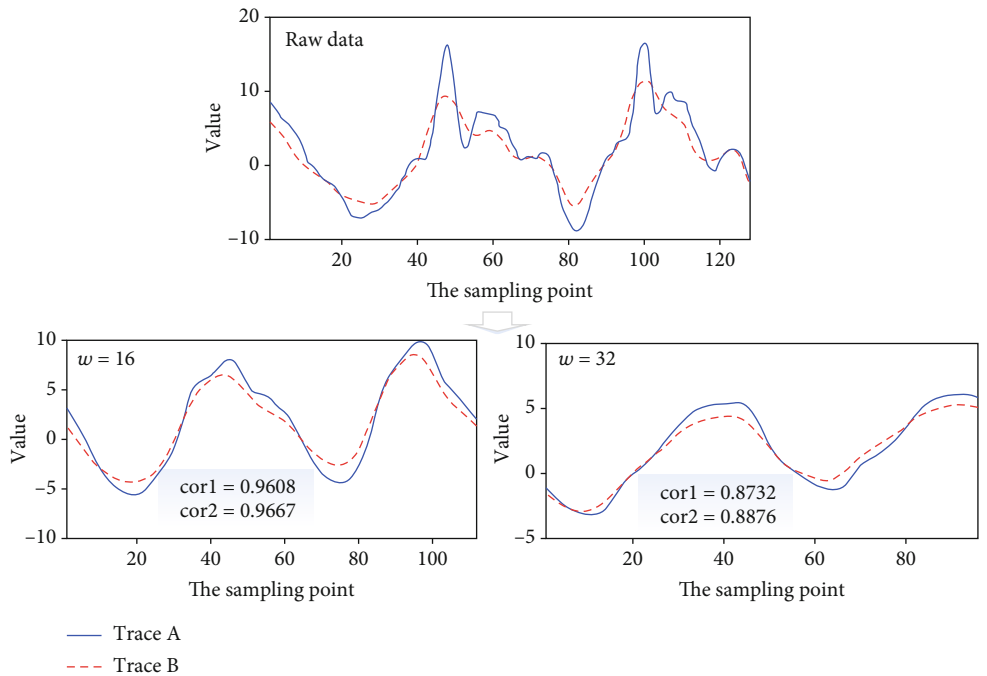


Figure 5: Interpolate the raw sensor data sequence.

the bit series, although we already take measures to reduce the possibility of mismatched bits in the previous steps. Therefore, we need an additional step to conduct mutual authentication and generate randomized and consistent keys.

Algorithm 4 shows the detailed design of mutual authentication and key extraction. In order to verify the consistency while preserving the privacy, it is essential for the peer devices to exchange a kind of signature for the generated bits to each other. Here, we leverage the hash-based message

Require: 1) The calibrated sensor data sequence with multiple dimension A_x, A_y, A_z and G_x, G_y, G_z: D_f; 2) The length of sensor data sequence: l.
Ensure: The bit series: S;
1: **for** each dimension i**do**
2: Conduct *Fast Fourier Transformation* over $D_{f,i}$ for dimension i. Compute the top-k principal components in the frequency domain, obtain the highest frequency of these components as f_i.
3: Sample the sensor data $D_{f,i}$ for every $l/2f_i$ points, use a sliding window W_i to get the maximum value and the minimum value, normalize the sampled data into the interval [-1,+1].
4: Use the level crossing method to quantize the data sequence $D_{f,i}$, obtain the generated bit series as S_i.
5: For each dimension, break the generated bit series S_i into blocks with variable size according to the specified time line. Merge these blocks from multiple dimensions to bit series S in time order.

Algorithm 3: Quantization.

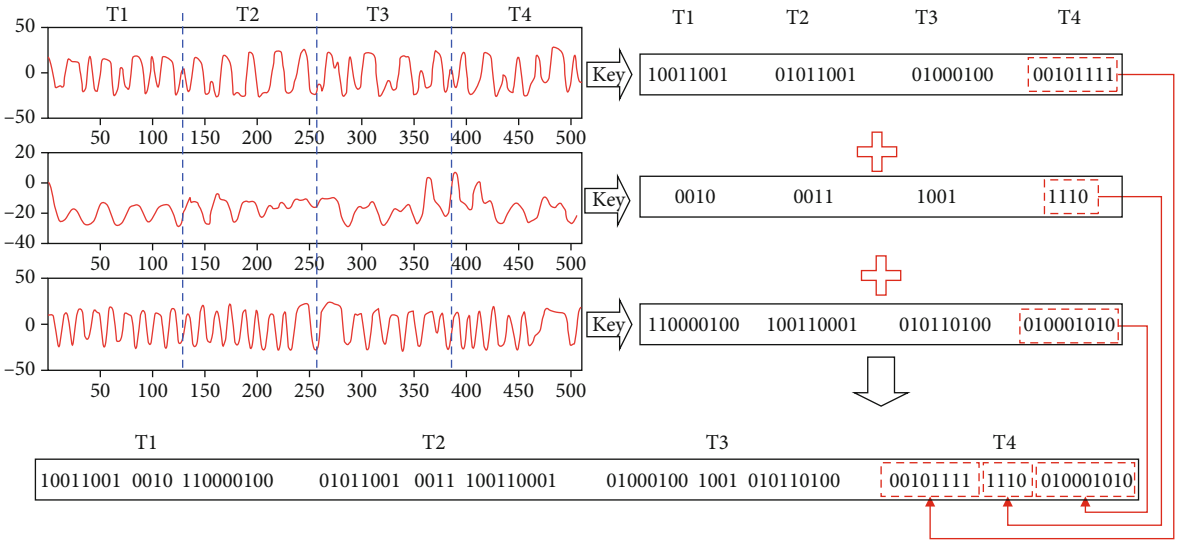


Figure 6: Quantization from different dimensions based on different sensitivities.

Require: 1) The bit series: S, its length is l. 2) The number of bits required in key generation: k. 3) The threshold for mutual authentication: t.
Ensure: The extracted key: K.
1: Initialize a sliding window W of size w along the bit series S. Set the index $i = 1$.
2: **while**$i \le l$**do**
3: Obtain the bit string inside window W as M_i, generate a random number r_i, and call HMAC(r_i, M_i) to obtain a bit string B_i.
4: Forward the window along the bit series S with 1 bit.
5: Send the series $X = \{r_i \| B_i\}(1 \le i \le l)$ with length l_X to the peer device via broadcasting.
6: Receive the series $Y = \{r'_i \| B'_i\}(1 \le i \le l)$ with length l_Y from the peer device.
7: Compute the edit distance d between X and Y, compute $\alpha = d/\max\{l_X, l_Y\}$, find the matching components of B_i and B'_i from X and Y accordingly.
8: **if**$\alpha \le t$**then**
9: Authenticate the peer device as an authorized device.
10: Find the first matching components of B_i and B'_i from X and Y, verify the consistency between the locally generated bits S with the bits S' from the peer device based on the following matching results.
11: Compute the edit distance between S and S', find the matching bits and divide them into multiple blocks, calculate the hash code M' for each block M, i.e., M'=HMAC(M), assemble them as a bit series, and extract the first k bits into the key K.

Algorithm 4: Mutual authentication and key extraction.

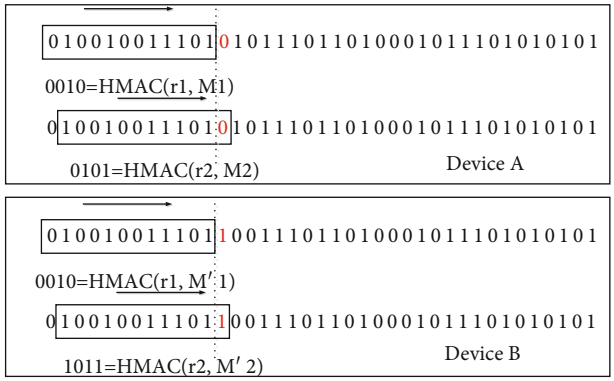


Figure 7: Decoding the exact bits for the peer device.

authentication code (HMAC) as the signature. Each device forwards a sliding window along the bit series S and calculates its HMAC on the bit series inside the window with a random number. For every step of 1 bit, each device calculates a new HMAC for the window. After that, each device exchanges the series of HMAC to the peer device. After receiving the HMAC series from the peer device, each device compares the *edit distance* [33] between the locally generated HMAC series and the received HMAC series. If the edit distance is below a specified threshold, then the peer device is authenticated as an authorized device; the mutual authentication is finished.

After that, according to the first matched HMAC and the following HMACs in the HMAC series, each device can locally figure out the exact bits of the following generated bits from the peer device. Figure 7 shows an example of decoding the exact bits for the peer device. Note that once device B gets the random number r_1 from device A, it calculates HMAC(r_1, M'_1) over the bits M'_1 inside its local window. Since it obtains the same value 0010 as A, it implies that the two bit series M_1 and M'_1 are equal. Then, it forwards the window with 1 bit and further verifies the HMAC with A; this time, M_2 is not equal to M'_2 since they do not have the same HMAC. Based on the previous result, it can be implied that the only difference lies in the exact 1 bit; device B can thus infer the exact bits in M_2. In this way, all bits from the peer device that follow the first matched bits can be effectively figured out. Note that, while exchanging the HMAC series between the peer devices, the attacker may intercept the message and try to solve M_i from HMAC(r_i, M_i) using a brute force searching. However, as long as we use a large enough window size w, e.g., w is larger than 64 bits, the user may have little chance to crack it.

At last, each device can calculate the *edit distance* [33] between the locally generated bits S and the bits S' from the peer device. Based on this, the series of matching bits can be effectively identified, even if the two matching bit series have some time lag in bit generation. This can effectively solve the out-of-sync problem during the shaking process, since one device may generate the same bits with a little time delay after the other. In order to reduce the correlations among the bit series, we further divide the matching bits into multiple blocks, compute HMAC for each block, and assemble them together as the final key. In this way, since the generated bits from each block all reflect the details of the user behavior, they always have some difference in both the time and space domains; as each bit in the output of HMAC is dependent on all bits in the input bit series, the generated bits can be fully randomized.

4.8. Security Analysis. In *eavesdropping attack*, attackers can launch eavesdropping attack by intercepting the transmitted messages in the open environment. Since the symmetric keys are generated independently among various devices and the messages broadcasted in key generation phase are only digital signatures used for verifying consistency, the attackers can never capture the actual value of the generated symmetric key.

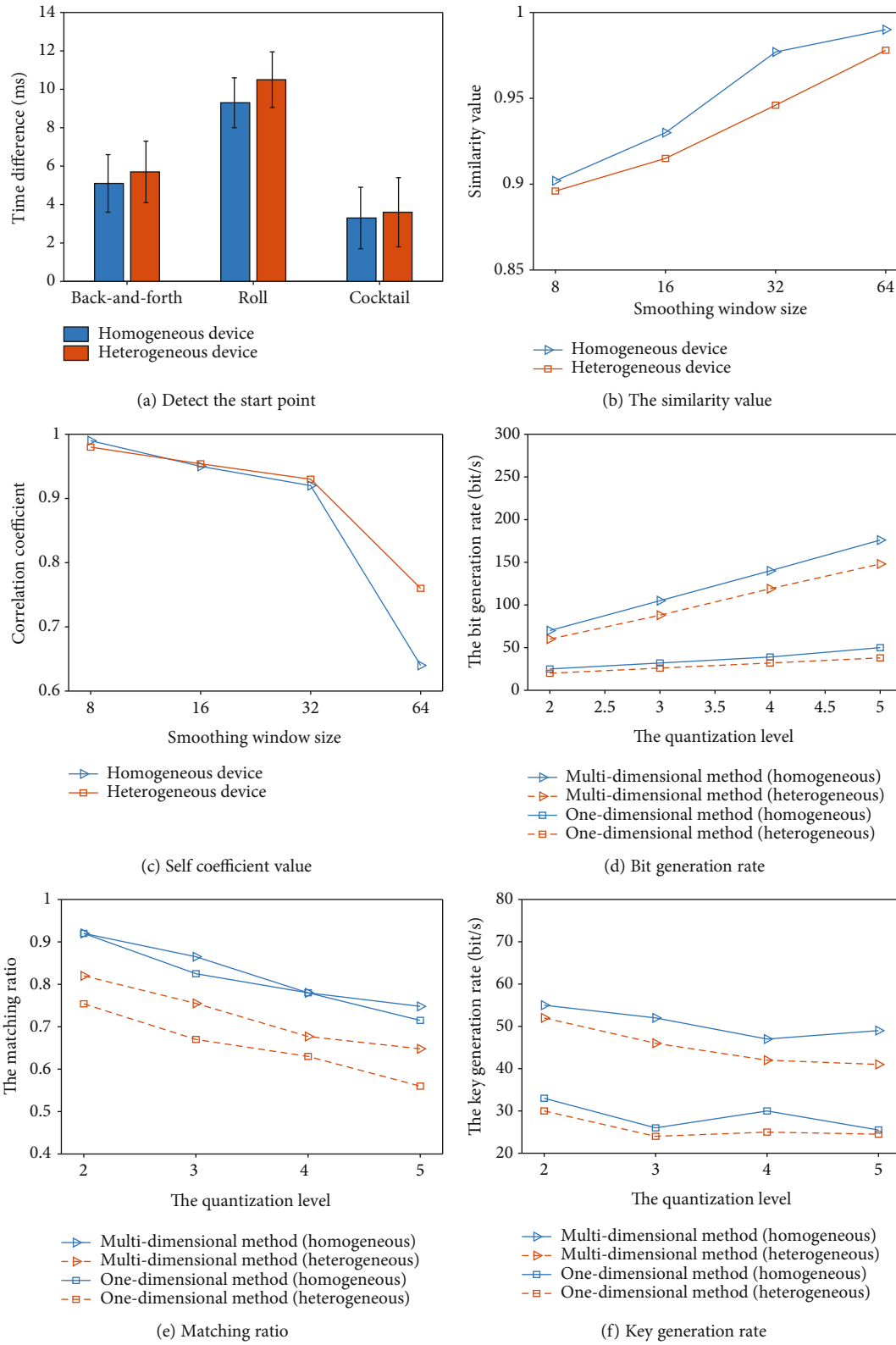
In *man-in-the-middle attack*, attackers can launch man-in-the-middle attack between two devices in interaction. As all participating devices are communicating in the open wireless channel, any man-in-the-middle attack will be overheard and detected by the legal devices; these devices can then actively reject the replayed fake messages.

In *shoulder surfing*, attackers can conduct shoulder surfing by privately imitating the legal user's behaviors in handshaking. However, as long as any two devices are not shaking together, there will exist differences in the detailed profile and pattern of user behaviors. Our solution can accurately identify the slight difference to prevent generating the same symmetric keys.

5. Performance Evaluation

5.1. Experiment Settings. We implemented our authentication architecture on an Android 4.0 operating system. We let 14 users (6 males and 8 females), respectively, shake two homogeneous and heterogeneous devices 10 times using three patterns (back and forth, roll, cocktail). So we get $14 \times 10 \times 3$ sets of sensor data. Besides, we also let one user imitate shaking behavior of another user using homogeneous/heterogeneous devices. There are in total 20 pairs of users that will have to do the experiments, and each pair of users does the same experiment 10 times. So we get another $20 \times 10 \times 3$ sets of sensor data. We adopt WiFi as the insecure wireless channel. In the start of authentication, each device sends the beacon to scan the adjacent devices and connects to each other.

5.2. Evaluate the Accuracy of Trace Synchronization. In data calibration, each device independently detects the start point of sampling. In order to show the performance of the trace synchronization algorithm, each device sends a beacon to the server as long as it detects the start point. Once receiving two beacons, the server calculates the time difference. Figure 8(a) shows the time difference of three patterns. The time difference of three patterns are all lower than 14 ms, and the average is about 8 ms. As the sampling frequency of devices is about 50-100 Hz, so the data on two shaking devices has just few point offsets, which can be removed in data interpolation. Besides, the time difference of homogeneous smart device is smaller than the one of the



(a) Detect the start point

(b) The similarity value

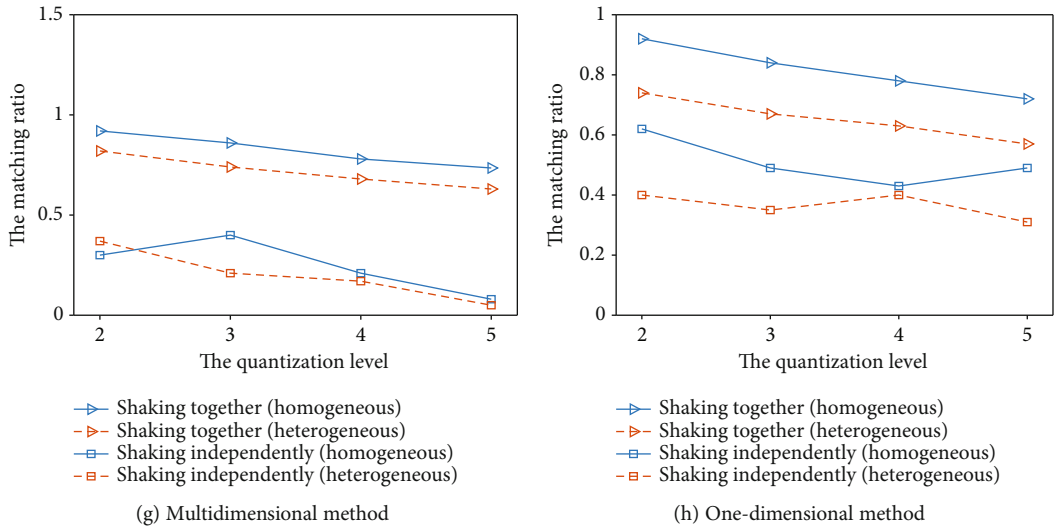
(c) Self coefficient value

(d) Bit generation rate

(e) Matching ratio

(f) Key generation rate

Figure 8: Continued.



(g) Multidimensional method (h) One-dimensional method

Figure 8: Performance evaluation.

heterogeneous device. This difference is caused by the sensor accuracy of the heterogeneous device. Overall, our algorithm could accurately detect the start sampling point.

5.3. Evaluate the Performance of Data Interpolation. In data interpolation, each device dynamically determines the smoothing window size w with the crosscorrelation coefficient t. Figure 8(b) shows the similarity of two sets of sensor data sampled from two shaking devices with different w. We can notice that as w increases, the similarity value increases, which means data interpolation effectively reduces the difference between sensor data by smoothing. Figure 8(c) shows the coefficient value of the sensor data with its own raw data with different w. We note that as w increases, the coefficient decreases. But the larger smoothing window size may remove the detail of the raw data. Fortunately, there is always a rapid decline of the coefficient value as w increases. So we can set t to a proper value, e.g., 0.85, to detect this change and this method works fine.

5.4. Evaluate the Ability of Key Generation. Our authentication architecture adopts six-dimensional sensor data instead of the amplitude of the sensor data. In order to compare the performance of those two methods, we use three metrics to evaluate the performance of our authentication architecture: *bit generation rate*, *matching ratio*, and *key generation rate*. Bit generation rate refers to the number of bits generated per second in the phase of quantization. Matching ratio refers to the percentage of the equal bits generated from two sets of sensor data in the phase of quantization. Key generation rate refers to the number of keys generated per second after key extraction. For simplicity, we represent the method of adopting six-dimensional sensor data as multidimensional method and the method of adopting the amplitude of the sensor data as one-dimensional method. Figure 8(d) shows the bit generation rate of two methods. We can see that the bit generation rate of the multidimensional method is far greater than the one of one-dimensional method. Besides, the bit generation rate of data sampled by homogeneous device is always greater than the case of heterogeneous device. Figure 8(e) shows the matching ratio of two methods. Those two methods have the similar matching ratio, which means that the multidimensional method could generate more bits than the one-dimensional method using an equal set of sensor data. That is because the multidimensional method effectively utilizes each dimension of the data while removing the differences. Figure 8(f) shows the comparison between the key generation rate of two methods. Like the previous case, the multidimensional method generates more keys than the one-dimensional method.

5.5. Evaluate the Imitation Resistance. In order to show the antiattack capability of our authentication architecture, we let one user imitate the shaking behavior of another user. We set the quantization level as 2. Figure 8(g) shows the matching rate of bits separately generated by the data sampled from the devices shaken together and independently using the multidimensional method. Figure 8(h) shows the same value using the one-dimensional method. We notice that even when the devices are not shaken together, both two devices could generate a certain amount of equal bits. However, the success rate by shaking devices together is all greater than the success rate by imitating. So each device could preset a threshold value, e.g., 0.7, to remove the false-positive situation. Besides, comparing these two figures, we can find that the discrimination between shaking together and shaking independently by the multidimensional method is more larger than the one by the one-dimensional method. So our authentication architecture can effectively detect the imitation behavior with homogeneous and heterogeneous devices.

5.6. Evaluate the Randomness via NIST Test. In order to verify the randomness of the final key, we perform the NIST tests on the extracted keys. We concatenate the generated keys together as the input of the NIST test. Among the 15

Table 1: NIST statistical test suite result.

NIST test	Homogeneous devices	Heterogeneous devices
Freq.	0.5004	0.4339
Block Freq.	0.8843	0.8596
Cum. sums (Fwd)	0.4506	0.42485
Cum. sums (Rev)	0.7511	0.5945
Runs	0.3426	0.4605
Longest run os ls	0.7187	0.6342
FFT	0.5629	0.5856
Approx. entropy	0.9365	0.9201
Serial	0.9702/0.9991	0.8053/0.7230

tests in the NIST tool, we run 8 tests. As shown in Table 1, the extracted bits can pass these tests as all the values are greater than 0.01. So the key generated from our authentication architecture is secure.

6. Limitations and Discussions

6.1. Robustness to Tackle the Interference from Ambient Environment. The issues like the interference from ambient environment, e.g., inconsistent movements among the shaking devices, could lead to the generation of mismatched bits in the bit series among the peer devices in secure paring. To tackle these issues, first, we take measures to reduce the possibility of mismatched bits in the data calibration, filtering, and quantization. Moreover, we leverage the hash-based message authentication code (HMAC) as the signature to generate consistent keys. After exchange of the HMAC series among the peer devices, each device can locally figure out the exact bits in the generated key in a consistent manner.

6.2. Generalization to Tackle the Differences for Various Sensors. iShake uses the sensor readings of the accelerometer and gyroscope in 3 dimensions to generate the randomized key. If there exist any differences for various sensors and dimensions, in regard to the sensitivity, sampling rate, and similarity, it could lead to the performance degradation in mutual authentication and key generation. To tackle these issues, we actually already considered these differences in the data calibration, filtering, and quantization and take measures to mitigate the impact from these differences. In the final key generation phase, we can further use the HMAC signature to remove the inconsistent bits caused by the differences for various sensors.

6.3. Cost versus Benefit among Different Secure Pairing Solutions. iShake uses the inertial measurements to perform secure pairing, including mutual authentication and key generation among peer devices. Since inertial measurement unit (IMU) has been brought to standard configuration for most state-of-art smart devices, it is both low in cost and pervasive in use. Compared to the more advanced secure pairing solutions like the near-field communication (NFC), the IMU-based solution in iShake achieves high cost performance ratio and pervasive usage.

7. Conclusion

In this paper, we propose iShake, a mutual authentication and key generation framework for smart devices, by shaking these devices together. As compared to previous work, iShake generates highly distinctive and fully randomized key between devices, and the whole process is more resistant to imitation. The main contributions are as follows: (1) we conduct comprehensive experiments in real environment, illustrate novel observations, and extract important clues for efficient key generation; (2) we propose a series of novel techniques to make the generated key highly distinctive and fully randomized; and (3) we are the first to consider key generation among heterogeneous devices and address new challenging issues in generating unique and consistent keys.

Data Availability

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

Disclosure

An earlier version of the manuscript has been presented at the IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS) in 2019. This work is an extended version of the earlier manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant Nos. 61872174, 61832008, and 61902175 and Jiangsu Natural Science Foundation under Grant No. BK20190293. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] L. Li, X. Zhao, and G. Xue, "A lightweight system to authenticate smartphones in the near field without NFC chips," in *2013 IEEE International Conference on Communications (ICC)*, pp. 6391–6395, Budapest, Hungary, 2013.

[2] L. Li, X. Zhao, and G. Xue, "Near field authentication for smart devices," in *2013 Proceedings IEEE INFOCOM*, pp. 375–379, Turin, Italy, 2013.

[3] M. Sethi, M. Antikainen, and T. Aura, "Commitment-based device pairing with synchronized drawing," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 181–189, Budapest, Hungary, 2014.

[4] M. Shahzad, A. X. Liu, and A. Samuel, "Behavior based human authentication on touch screen devices using gestures and signatures," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2726–2741, 2016.

[5] Y. Song, Z. Cai, and Z.-L. Zhang, "Multi-touch authentication using hand geometry and behavioral information," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 357–372, San Jose, CA, USA, 2017.

[6] J. Lester, B. Hannaford, and G. Borriello, "Are you with me?-using accelerometers to determine if two devices are carried by the same person," in *International Conference on Pervasive Computing*, pp. 33–50, Springer, 2004.

[7] R. Mayrhofer, "The candidate key protocol for generating secret shared keys from similar sensor data streams," in *European Workshop on Security in Ad-hoc and Sensor Networks*, pp. 1–15, Springer, 2007.

[8] R. Mayrhofer and H. Gellersen, "Shake well before use: authentication based on accelerometer data," in *International Conference on Pervasive Computing*, pp. 144–161, Springer, 2007.

[9] R. Mayrhofer and H. Gellersen, "Shake well before use: intuitive and secure pairing of mobile devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 792–806, 2009.

[10] R. Marin-Perianu, M. Marin-Perianu, P. Havinga, and H. Scholten, "Movement-based group awareness with wireless sensor networks," in *International Conference on Pervasive Computing*, pp. 298–315, Springer, 2007.

[11] K. Hinckley, "Synchronous gestures for multiple persons and computers," in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pp. 149–158, Vancouver, Canada, 2003.

[12] Y. Shen, F. Yang, B. Du, W. Xu, C. Luo, and H. Wen, "Shake-n-shack: enabling secure data exchange between smart wearables via handshakes," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, Athens, Greece, 2018.

[13] Q. Jiang, X. Huang, N. Zhang, K. Zhang, X. Ma, and J. Ma, "Shake to communicate: secure handshake acceleration-based pairing mechanism for wrist worn devices," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5618–5630, 2019.

[14] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, "Shakeunlock: securely transfer authentication states between mobile devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 1163–1175, 2016.

[15] S. Muhammad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it," in *Proceedings of the 19th annual international conference on Mobile computing & networking - MobiCom '13*, pp. 39–50, Miami, Florida, 2013.

[16] W. Xu, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, "Walkie-talkie: motion-assisted automatic key generation for secure on-body device communication," in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12, Vienna, Austria, 2016.

[17] S. Mare, R. Rawassizadeh, R. Peterson, and D. Kotz, "Continuous smartphone authentication using wristbands," in *Proceedings of the Workshop on Usable Security*, pp. 1–12, Stockholm, Sweden, 2019.

[18] H.-M. C. Leung, C.-W. Fu, and P.-A. Heng, "TwistIn," *Proceedings of the ACM on Inter-active, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, pp. 1–24, 2018.

[19] S. Mare, R. Rawassizadeh, R. Peterson, and D. Kotz, "SAW: wristband-based authentication for desktop computers," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–29, 2018.

[20] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proceedings of the 15th annual international conference on Mobile computing and networking - MobiCom '09*, pp. 321–332, Beijing, China, 2009.

[21] X. Zhu, F. Xu, E. Novak, C. C. Tan, Q. Li, and G. Chen, "Extracting secret key from wireless link dynamics in vehicular environments," in *2013 Proceedings IEEE INFOCOM*, pp. 2283–2291, Turin, Italy, 2013.

[22] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *2012 Proceedings IEEE INFOCOM*, pp. 927–935, Orlando, FL, USA, 2012.

[23] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *2013 Proceedings IEEE INFOCOM*, pp. 3048–3056, Turin, Italy, 2013.

[24] C. Castelluccia and P. Mutaf, "Shake them up!: a movement-based pairing protocol for CPU-constrained devices," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pp. 51–64, Seattle, Washington, 2005.

[25] T. Zhang, X. Yi, R. Wang et al., "Tap-to-pair: associating wireless devices with synchronous tapping," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–21, 2018.

[26] S. Mei, Z. Liu, Y. Zeng, L. Yang, and J. F. Ma, "Listen!: audio-based smart IoT device pairing protocol," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*,, pp. 391–397, Xi'an, China, 2019.

[27] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: a technique for users to easily establish connections between smart artefacts," in *International Conference on Ubiquitous Computing*, G. D. Abowd, B. Brumitt, and S. Shafer, Eds., pp. 116–122, Springer, 2001.

[28] W. Xu, C. Javali, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, "Gait-Key: a gait-based shared secret key generation protocol for wearable devices," *ACM Transactions on Sensor Networks*, vol. 13, no. 1, pp. 1–27, 2017.

[29] G. Revadigar, C. Javali, W. Xu, A. V. Vasilakos, W. Hu, and S. Jha, "Accelerometer and fuzzy vault-based secure group key generation and sharing protocol for smart wearables," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2467–2482, 2017.

[30] D. Bichler, G. Stromberg, M. Huemer, and M. Low, "Key generation based on acceleration data of shaking processes," in *International Conference on Ubiquitous Computing*, pp. 304–317, Springer, 2007.

[31] D. Bichler, G. Stromberg, and M. Huemer, "Synchronizing shaking sequences for generating symmetric keys," in *2009 2nd International Workshop on Nonlinear Dynamics and Synchronization*, pp. 75–80, Klagenfurt, Austria, 2009.

[32] Y. Shen, B. Du, W. Xu et al., "Securing cyber-physical social interactions on wrist-worn devices," *ACM Transactions on Sensor Networks*, vol. 16, no. 2, pp. 1–22, 2020.

[33] E. Ukkonen, "On approximate string matching," in *Foundations of Computation Theory*, pp. 487–495, Springer, 1983.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6648775, 11 pages
https://doi.org/10.1155/2021/6648775

Research Article

A Sampling-Based Method for Highly Efficient Privacy-Preserving Data Publication

Guoming Lu,[1,2] **Xu Zheng**,[1,2] **Jingyuan Duan**,[1] **Ling Tian**,[1,2] **and Xia Wang**[3]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Trusted Cloud Computing and Big Data Key Laboratory of Sichuan Province, Chengdu 610000, China*
[3]*School of Statistics and Data Science, Beijing University of Technology, Beijing, China*

Correspondence should be addressed to Xu Zheng; xzheng@uestc.edu.cn

Received 26 October 2020; Revised 19 February 2021; Accepted 3 March 2021; Published 15 March 2021

Academic Editor: Lin Wang

The data publication from multiple contributors has been long considered a fundamental task for data processing in various domains. It has been treated as one prominent prerequisite for enabling AI techniques in wireless networks. With the emergence of diversified smart devices and applications, data held by individuals becomes more pervasive and nontrivial for publication. First, the data are more private and sensitive, as they cover every aspect of daily life, from the incoming data to the fitness data. Second, the publication of such data is also bandwidth-consuming, as they are likely to be stored on mobile devices. The local differential privacy has been considered a novel paradigm for such distributed data publication. However, existing works mostly request the encoding of contents into vector space for publication, which is still costly in network resources. Therefore, this work proposes a novel framework for highly efficient privacy-preserving data publication. Specifically, two sampling-based algorithms are proposed for the histogram publication, which is an important statistic for data analysis. The first algorithm applies a bit-level sampling strategy to both reduce the overall bandwidth and balance the cost among contributors. The second algorithm allows consumers to adjust their focus on different intervals and can properly allocate the sampling ratios to optimize the overall performance. Both the analysis and the validation of real-world data traces have demonstrated the advancement of our work.

1. Introduction

Enabling the pervasive adoption of cutting-edge techniques of artificial intelligence usually requests the support of huge scales of data [1]. With the joint contribution of smart devices and easy network access, the emergence of volumes of data has been extended from those dominating enterprises to individual contributors like IoT devices. Therefore, the collection of contents in wireless manners has been considered a fundamental task for data processing and AI enhancing [2]. However, the concerns on privacy and resource consumption also rise accordingly. The ubiquitous availability of data sources has broken the boundary between cyber life and physical life. It is believed that every aspect of our life is recorded by some data, thus providing numerous challenges for privacy-preserving data sharing [3]. Meanwhile, data are usually uploaded via wireless or wired networks as they are stored on personal devices. Then, the publication of data is also resource-consuming, especially for those multimedia contents. Both factors have significantly hindered the pervasive collection and make it a nontrivial problem. Therefore, this paper proposes a sampling-based strategy for private data publication.

Actually, data consumers can accept or are even more interested in the statistics about data instead of the detailed contents from every contributor. Such statistics could be sufficient and more reliable for analysis and decision-making. For example, service providers may apply the scale of traffic loads for traffic prediction, while regular users can plan their routes based on such statistics [4]. Among these statistics, the histogram, which provides the distribution of underlying facts, is believed to be essential for data analysis [5]. It may act as an index showing the portions of users falling in the category. Meanwhile, the histogram also provides sights for

privacy preservation, as individuals do not disclose their original contents to data brokers or consumers.

To formalize such insights, the local differential privacy [6] is proposed and considered to be a novel paradigm for privacy preservation under distributed manners. LDP is extended from the original differential privacy by removing the request of a trustable data curator. In a typical LDP framework, each participant locally holds her contents and reports encoded and obfuscated results to the data broker, which will aggregate the contents to generate statistics. In this way, the individual contents are preserved, while the noise can be reduced during aggregation. Existing works are conducted for different types of statistics. For categorical values, the heavy hitters, frequent itemsets, and many other statistics are investigated [7]. For numerical values, the mean value, the summation, and some other aggregation queries are studied [8]. Meanwhile, there is also a batch of studies focusing on efficient and fair data publication under various scenarios [9–11].

However, current works for LDP request the encoding of original contents, which are usually bandwidth-consuming. Take the random response as an example [12]. It encodes the value into K-folds, where each fold represents one category of value, like the visited website. As all K-folds will be encoded, the bandwidth for encoded contents will be huge. This is extremely difficult when the candidates of values are large or even infinite (i.e., numerical value). Although some works are conducted, they have not thoroughly considered the problem.

Fortunately, the data consumers, when dealing with statistics, can actually accept minor uncertainty in the results. This is due to the inherent bias underlying the collected data, where the contributors themselves are also samples of the whole population. Meanwhile, statistics with bounded minor errors will not affect decision-making. Therefore, it is interesting to study whether we can further reduce bandwidth consumption while maintaining utility and privacy preservation.

As a result, this work proposed a novel framework for the publication of histograms in distributed manners. In the framework, data contributors locally hold their contents, like their incoming data. The data consumers request histograms with different granularities. The data brokers act as the coordinators among them, which are assumed to be semihonest. They will collect the results from participants and try to infer the raw contents beneath them. A sampling-based algorithm is designed, where the raw data are first encoded with randomized perturbation, and then a bit-level sample strategy is applied for publication. The data brokers will decode the sampled results and respond to consumers with aggregated histograms.

In the framework, all participants are assured with local differential privacy, which is theoretically proved under the sampling strategy. Furthermore, we also propose a mechanism where participants can efficiently derive the encoding scheme under multiple histograms with heterogeneous queries. As for the sampling, two strategies are given, based on whether queries address the same focus on all intervals. We prove the unbiased results for the first sampling and the optimized allocation of bandwidth under the second strategy. Finally, we evaluate the performance under real-world datasets. The results demonstrate the efficiency of our methods. As far as we know, this is the first study on the sample-based histogram publication over numerical values under local differential privacy. The main contribution of this work includes the following:

(i) A novel framework for efficient and privacy-preserved histogram publication over multiple participants

(ii) Two sampling-based strategies for distributed histogram publication under LDP

(iii) Theoretical analysis on the accuracy, efficiency, and privacy preservation

(iv) Evaluation on real-world data traces to demonstrate the effectiveness of proposed methods

The rest of the paper is organized as follows. Section 2 reviews the literature works. Section 3 proposes the problem formulation and some preliminaries. Section 4 introduces sampling-based algorithms for histogram publication. The evaluation results are shown in Section 5. Section 6 concludes the paper.

2. Related Work

2.1. Privacy-Preserved Data Publication. The publication of private data has been extensively studied during the past decades. Typical techniques including K-anonymity are proposed [13–16], where the sensitive contents are mixed and obfuscated before publication. However, these studies usually request the limited background knowledge of adversaries and are vulnerable to specific types of attacks. The differential privacy, as a novel index for privacy preservation, allows the existence of arbitrary attackers. There are also some studies investigating histogram publication under differential privacy, focusing on different types of data [17, 18]. They also apply the underlying properties of these data to further reduce the degrees of noise [5, 19, 20]. However, they assume the existence of a trustable data curator to coordinate the data publication, which is usually infeasible for distributed data publication. Finally, there are also some studies considering the fairness and other issues within the private data publication [21–23]. However, they fail to properly reduce the bandwidth consumption and are not compatible with the histogram publication.

2.2. Local Differential Privacy. Local differential privacy [6] provides a novel paradigm for distributed data publication under differential privacy. It allows multiple data contributors to privately aggregate their contents when the data broker is semihonest. Multiple types of statistics are studied, including the publication of graph structures [24], the range counting [25], and the histogram distribution [26]. There are also some studies investigating the efficiency of data publication, ranging from the RAPPOR and Basic RAPPOR methods [12] proposed by Google to sophisticated methods

where more mathematical solutions are applied. Reference [27] reviews current studies on LDP providing guidelines for applications. Meanwhile, there is also a batch of studies on the publication of numerical values, and statistics like mean values are considered [8, 28, 29]. However, these studies tend to encode the numerical value into several fixed values, and the perturbed contents may fall out of the original range. This may reduce the utility of published contents [30, 31]. Therefore, the design of an efficient mechanism for histogram publication under local differential privacy is still a challenging topic.

2.3. Sampling-Based Data Collection. Finally, the sampling-based strategy has also been studied for data collection from multiple contributors. Maybe one typical domain where the sampling strategy functions well is the Internet of Things. The wireless and battery-powered sensors and actuators are usually limited in resources. The sampling-based data collection can balance the accuracy of the results and the devoted resources. Corresponding studies are conducted on statistics like Top-K values and data sketching.

As for the combination of sampling strategies and privacy preservation, there are also some studies arguing that the sampling component can strengthen the degree of differential privacy. However, these studies request content-level sampling, which means the sampled contributors can save no resources. Applying the sampling strategy while flexibly balancing the bandwidth consumption is still not well addressed.

3. Problem Formulation

This section first provides the problem formulation for distributed histogram publication, including the network and attacking models. Then, preliminaries on local differential privacy are given.

3.1. Network Model. The whole platform consists of three parties: the data brokers, the data consumers, and data contributors. Initially, data consumers post their histogram queries to data brokers, denoted as $l_1, l_2, \cdots, l_M$. To simplify our model, l_i indicates both the ith query and the interval length of the histogram requested by the query, i.e., the granularity of the histogram. Different consumers may request different diverse queries with different granularities, as they can hold heterogeneous purposes. For example, taxi companies request a coarsened level of traffic loads to guide the deployment of their services, while the navigation apps expect fine-grained histograms to generate a fast route. Upon receiving the requests, the data brokers will generate a data collection plan among participants. The plan consists of a set of consecutive intervals $[D_0, D_1), [D_1, D_2), \cdots, [D_K - 1, D_K]$, together with other parameters like the sampling ratio. As for the intervals, $D_0 = D_L$ indicates the minimum value of all contents, while $D_K = D_U$ refers to the maximum value of all contents. In this framework, we will focus on the snapshot query, such that the data brokers will collect queries from all consumers before generating a plan. Therefore, the intervals are generated based on all queries.

By receiving the plan from data brokers, data contributors will encode and report their contents. Assume N contributors exist in the system, noted as $\{u_1, u_2, \cdots, u_N\}$. Each of them holds one content d_i, and $D_L \le d_i \le D_U$. All contents belong to one dimension or can be applied for the same type of query, like the humidity level of a local area or the various sensing data capturing the traffic congestion. We assume the total bandwidth used for reporting d_i to the data brokers as B_i.

Finally, the data broker collects results from all contributors. It will first decode the reported contents and then aggregate them into different intervals. In a final step, the data broker will generate and distribute corresponding histograms to different consumers and charge them accordingly.

3.1.1. Adversarial Model. Due to the latent value of contents held by contributors, both data brokers and consumers are assumed to be semihonest. It means they will not break into contributors' devices to steal the contents but will follow the framework and try to infer the original contents. Therefore, contributors should carefully encode their contents to thwart such inference attacks. The local differential privacy has been considered an extension of differential privacy by removing the requirement of a trusted data broker. LDP still allows arbitrary background knowledge from the adversaries and can preserve individual contents within the statistics. To achieve the LDP property, data contributors can publish perturbed contents d_i, which are either noise values or some relative data structures. The definition of local differential privacy is shown in Definition 1.

Definition 1 (local differential privacy). An algorithm $Q(\cdot)$ satisfies ε-local differential privacy (ε-LDP), where $\varepsilon \ge 0$, if and only if for two arbitrary contents T_i and T_j:

$$\forall y \in \text{Range}(Q): \ \Pr[Q(T_i) = y] \le e^{\varepsilon} \Pr\left[Q\left(T_j\right) = y\right], \quad (1)$$

where $\text{Range}(Q)$ denotes the set of all possible outputs of $Q(\cdot)$.

Based on its definition, the local differential privacy ensures that no significant information will be disclosed to the data receivers. The parameter ε indicates the degree of privacy, where a larger ε means data contributors are less sensitive and will produce more accurate results.

3.1.2. Design Object. Based on the network model and attacks from the adversaries, data contributors aim to both reduce their bandwidth consumption and preserve their raw contents during data uploading. The data brokers are concerned with coordinating the trading between other parties, so their focus is to generate a proper plan for data collection. The plan should be efficient and provide rational performance. Generally, assume that the accumulated variance for each query l_i is $\text{Var}(\text{l}_\text{i})$. The design object is given as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{M} l_i \\ \text{s.t.} \quad & E\left(R_{jk}{}'\right) = R_{jk}, \quad \forall l_1, l_2, \cdots, l_M \\ & \sum B_i \le B_0 \\ & d_i \text{ is preserved under LDP}, \quad \forall i \in \{1, 2, \cdots, N\}, \end{aligned} \tag{2}$$

which means the derived result should be unbiased, the total bandwidth should be constrained, and the privacy for each contributor should be preserved.

3.2. Preliminaries. This part introduces preliminaries for LDP. It first introduces one basic encoding-decoding-based method for data uploading and then addresses the compositional properties of LDP.

The random response method provides some basic ideas for the implementation of LDP. We take the Basic RAPPOR proposed by Google as an example.

In Basic RAPPOR, assume there is a L-bit vector with binary entry, denoted as $V = (v_1, v_2, \cdots, v_L)$. $v_i = 1$ indicates that the data item d belongs to the ith category; otherwise, $v_i = 0$.

Then, V' can be generated by the randomized response:

$$\Pr\left[V'[i] = 1\right] = \begin{cases} 1 - \frac{1}{2}f, & \text{if } V[i] = 1, \\ \frac{1}{2}f, & \text{if } V[i] = 0. \end{cases} \tag{3}$$

Finally, V' will be sent to the data curator for subsequent analysis. Actually, this mechanism of perturbation achieves the LDP property for vector V, which is proved by a previous study [27].

Theorem 2. *For an arbitrary vector $V = (v_1, v_2, \cdots, v_L)$, the Basic RAPPOR achieves ε-LDP for $\varepsilon = \ln\left(((1 - 1/2f)/(1/2f))^2\right)$.*

Data sampling, where contributors only partially upload their contents, is also a major strategy for resource-saving in distributed data collection. It is believed that this can further reduce the disclosure of information. Li et al. have theoretically proved the effect [32].

Theorem 3. *Assume $F(\cdot)$ to be an ε-differentially private algorithm and $S(\cdot)$ to be a sampling method algorithm. Then, if $S(\cdot)$ is first applied to a dataset, which is later perturbed by $F(\cdot)$, the derived result satisfies $\ln(1 + P_0(e^{\varepsilon} - 1))$-differential privacy, where P_0 is the sampling probability.*

Finally, the compositional property of differential privacy can also be merged with the LDP.

Theorem 4 (differentially sequential composition). *Let $\{F_1(\cdot), F_2(\cdot), \cdots, F_k(\cdot)\}$ be a set of functions satisfying differential privacy and privacy budgets be $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_k$, respectively. Then, applying all $F_i(\cdot)$ to one data item d_0 will provide $\sum_{i=1}^{k}\varepsilon_i$-differential privacy.*

4. Sampling-Based Histogram Publication

This part first provides a scheme applied for plan generation. Then, we argue that the efficiency could be further improved among all contributors by uploading partial results. The second part gives a bit-level sampling algorithm, where the bandwidth consumption among contributors is reduced and balanced. The third part proposes a biased sampling mechanism, where the sampling ratios can be adjusted and optimized according to the requests of consumers.

4.1. Data Plan for Multiple Queries. Within the framework, the data broker receives multiple queries $\{l_1, l_2, \cdots, l_M\}$ from consumers. It will then generate a corresponding plan $[D_0, D_1), [D_1, D_2), \cdots, [D_K - 1, D_K]$ from these queries. The main objective is to reduce the number of intervals in the plan, as each interval may reflect extra bandwidth consumption. The data broker applies the following strategy for plan generation.

The data broker iteratively generates intervals according to each query, which is determined by

$$[D_0, D_0 + l_i \cdot 1), [D_0 + l_i \cdot 1, D_0 + l_i \cdot 2), \cdots, [D_0 + l_i \cdot K_i, D_K], \tag{4}$$

where

$$K_i = \left\lfloor \frac{D_U - D_L}{l_i} \right\rfloor. \tag{5}$$

Then, the data broker combines all intervals from all queries. Specifically, every two consecutive checkpoints $D_0 + l_i \cdot j$ and $D_0 + l_m \cdot n$ from one or two queries will compose a new interval $[D_0 + l_i \cdot j, D_0 + l_m \cdot n)$. The newly generated intervals will be used by the data plan, as $[D_0, D_1), [D_1, D_2), \cdots, [D_K - 1, D_K]$. Therefore, the histogram for each query can be derived from such intervals by iteratively merging the results in several conjunctive intervals. The whole procedure will take $O(M \cdot (D_U - D_L))$ time. In the baseline method, data contributors may follow the typical random response to locally encode and obfuscate their contents. The results will be uploaded to data curators, which will further decode the results and publish the aggregated histograms to consumers.

4.2. Bit-Level Sampling for Histogram Publication. This part introduces the sampling-based algorithm for histogram publication. Intuitively, the data curator can randomly pick a group of contributors for histogram publication, or the contributors can locally determine whether to participate in data processing with some sampling ratios. However, the sampled contributors have to apply the encoding mechanisms and fully upload the vectors in both cases, which is unbalanced and unwilling. Therefore, this part proposes an improved algorithm to sample the contents from another dimension.

The algorithm is named as Bit-Sampling Histogram Publication (BSHP for short).

The main idea of BSHP is to implement bit-level sampling among contributors instead of one-time participant selection. Initially, BSHP follows the same steps as the baseline method, where the data curator processes and distributes the queries to contributors. After locally encoding their data values, contributors follow the request of data collection from the data curator in BSHP. In the jth iteration, the data curator announces a sampling ratio P_j to all contributors u_i, where $0 \le P_j \le 1$. Then, all u_i locally execute Bernoulli sampling with probability P_j. Contributors who sampled 1 as the result will upload the corresponding bit $D_{ij}{}'$ to the data curator. This request phase repeats K_0 rounds until all bits are requested.

In the decoding phase, the data curator first estimates the counting of data values in each interval of the integrated partition, where

$$R_k = \frac{\left(\left\|\left\{D_{i'} \mid D'_{ik} = 1\right\}\right\| / p_k\right) - 1/2 \cdot f \cdot N}{1 - f}, \quad \forall k \le K_0. \tag{6}$$

Then, the data curator derives the results for different queries in the same way with the baseline method, and the whole algorithm terminates.

4.2.1. Analysis. This part analyzes the performance of BSHP. We first investigate the accuracy of histogram publication and then discuss issues on privacy preservation.

The following theorem indicates that BSHP provides an unbiased estimation for the counting of each interval in the integrated partition.

Theorem 5 (unbiased estimation). *For each interval* $[W_k, W_{k+1}]$*, BSHP can provide an unbiased estimation under given* P_k *and* ε_0*, indicating* $E(R_k) = R_k^0$.

Proof. Within BSHP, the data curator estimates R_k as

$$R_k = \frac{\left(\left\|\left\{D_{i'} \mid D'_{ik} = 1\right\}\right\| / p_k\right) - 1/2 \cdot f \cdot N}{1 - f}. \tag{7}$$

We denote R_k^0 as the total size of contents belonging to interval $[W_k, W_{k+1}]$. According to the definition of the random response, we have

$$R_k = R_k^0 \cdot V_s \cdot V_r + \left(N - R_k^0\right) \cdot V_s \cdot V_r{}', \tag{8}$$

where V_s refers to the sampling variable whether a contributor is selected, and V_r and $V_r{}'$ indicate whether the corresponding bit is retained or reversed.

We have the following notation:

$$\Phi = R_k - R_k^0. \tag{9}$$

Then,

$$\begin{aligned} E(\Phi) = E\left(R_k - R_k^0\right) &= E\left(\frac{\left\|\left\{D_{i'} \mid D'_{ik} = 1\right\}\right\| - N \cdot P_k \cdot 1/2f}{(1-f) \cdot P_k}\right) \\ &- R_k^0 = \frac{E\left(\left\|\left\{D_{i'} \mid D'_{ik} = 1\right\}\right\|\right) - N \cdot P_k \cdot 1/2f}{(1-f) \cdot P_k} \\ &- R_k^0 = \frac{E\left(R_k^0 V_s V_r + \left(N - R_k^0\right) V_s V_{r'}\right) - NP_k 1/2f}{(1-f) \cdot P_k} \\ &- R_k^0. \end{aligned} \tag{10}$$

As V_s, V_r, and $V_r{}'$ are independent variables,

$$\begin{aligned} &E\left(R_k^0 V_s V_r + \left(N - R_k^0\right) V_s V_{r'}\right) \\ &= R_k^0 E(V_s) E(V_r) + \left(N - R_k^0\right) E(V_s) E(V_{r'}) \\ &= R_k^0 P_k \left(1 - \frac{1}{2} f\right) + \left(N - R_k^0\right) P_k \frac{1}{2} f. \end{aligned} \tag{11}$$

Therefore,

$$E(\Phi) = \frac{R_k^0 P_k (1-f)}{(1-f) \cdot P_k} - R_k^0 = 0. \tag{12}$$

Then, we have

$$E(R_k) = R_k^0, \tag{13}$$

which means R_k is an unbiased estimator for the counting of data values in $[W_k, W_{k+1}]$.

According to Theorem 5, BSHP provides an unbiased estimation for each interval in the integrated partition. Therefore, the final outputs for each query will also be an unbiased estimation, as the final results are derived from the combination of these unbiased countings.

The variance of the estimated result for an interval is calculated in Lemma 6. The main idea of the lemma is to combine the variance from two steps of sampling and derive the correlation between the variance and the two parameters P_k and ε_0.

Lemma 6. *For each interval* $[W_k, W_{k+1}]$*, with parameters* P_k *and* ε_0*, the variance follows* $Var(R_k) \le (((R_k^0)^2 + N^2 1/2f)/((1-f)^2 P_k)) + (1/4)(Nf^2/(1-f)^2) + (2NR_k^0 f/(1-f)^2)$.

Proof. First of all,

$$\mathrm{Var}(\Phi) = \mathrm{Var}\left(R_k - R_k^0\right) = \mathrm{Var}(R_k). \tag{14}$$

Assume $R_k' = \|\{D_{i'} \mid D'_{ik} = 1\}\|$. Then,

$$\begin{aligned}\mathrm{Var}(\Phi) &= E(\Phi^2) - E(\Phi)^2 = E(\Phi^2) \\ &= E\left[\left(\frac{R_{k'} - NP_k 1/2f}{(1-f)\cdot P_k} - R_k^0\right)^2\right] \\ &= E\left[\left(\frac{R_{k'} - NP_k 1/2f}{(1-f)\cdot P_k} - R_k^0\right)^2\right] \\ &= \left(R_k^0\right)^2 + E\left(\frac{(R_{k'} - NP_k 1/2f)^2}{(1-f)^2\cdot P_k^2}\right) \\ &\quad - 2R_k^0 \cdot E\left(\frac{R_{k'} - NP_k 1/2f}{(1-f)\cdot P_k}\right).\end{aligned} \tag{15}$$

As V_s is the Bernoulli sampling,

$$E(V_s^2) = E(V_s). \tag{16}$$

The same conclusion also holds for V_r and V_r'. Therefore,

$$\begin{aligned}\mathrm{Var}(\Phi) &= \left(R_k^0\right)^2 + \frac{E(R_{k'} - NP_k 1/2f)^2}{(1-f)^2\cdot P_k^2} - 2R_k^0 \cdot R_k^0 \\ &= \frac{E\left((R_{k'})^2\right) - N^2 P_k^2 1/4f^2 - NP_k f E(R_{k'}))}{(1-f)^2\cdot P_k^2} - \left(R_k^0\right)^2 \\ &= \frac{NP_k f}{(1-f)^2\cdot P_k^2}\left(R_k^0 P_k\left(1-\frac{1}{2}f\right) + \left(N - R_k^0\right)P_k\left(\frac{1}{2}f\right)\right) \\ &\quad - \left(R_k^0\right)^2 - \frac{N^2 P_k^2 1/4f^2}{(1-f)^2\cdot P_k^2} + \frac{\left(R_k^0\right)^2 P_k(1-1/2f) + \left(N - R_k^0\right)^2 P_k 1/2f}{(1-f)^2\cdot P_k^2} \\ &\quad + \frac{R_k^0\left(N - R_k^0\right)P_k^2(1-1/2f)f}{(1-f)^2\cdot P_k^2} \le \frac{\left(R_k^0\right)^2 + N^2 1/2f}{(1-f)^2 P_k} + \frac{1}{4}\frac{Nf^2}{(1-f)^2} \\ &\quad + \frac{2NR_k^0 f}{(1-f)^2}.\end{aligned} \tag{17}$$

As for each data consumer, the variance of the histogram is determined by the summation of variances in different intervals. We omit the conclusion here as the summation is straightforward.

Now we analyze the property of privacy preservation by BSHP. In BSHP, each contributor will only upload partial results of their vectors. Meanwhile, the perturbation and sampling could actually be applied in any order. According to the conclusions in [32], the sampling will strengthen privacy preservation. Therefore, we have the following conclusion.

Theorem 7 (local differential privacy). *BSHP preserves the data value for each contributor under ε'-local differential privacy, where $\varepsilon' \le \varepsilon_0$.*

4.3. Weighted Sampling for Histogram Publication. This part further studies the sampling method for histogram publication. Specifically, the data consumers may hold different requests on histograms. Taking the incoming data as an example, some consumers may prefer more accurate results for people with high salaries, while others may expect to derive the results in the middle of the population. Therefore, the algorithms for histogram publication should also handle such sophisticated utilities for consumers. The proposed algorithm is named as Weighted-Sampling Histogram Publication (WSHP for short).

Initially, WSHP allows each consumer to report their weights at different intervals. The weights for all intervals in the ith query are

$$\left\{\omega_{i1}, \omega_{i2}, \cdots, \omega_{iK_i - 1}\right\}. \tag{18}$$

The data curator first derives the integrated partition on the whole range, i.e., $\{W_1, W_2, \cdots, W_{K_0}\}$. Then, WSHP counts the accumulated weights for each interval. For an arbitrary interval $[W_i, W_{i+1}]$, its weight ω_i is derived by adding up corresponding weights from all contributors. Assume the jth query has its kth interval covering $[W_i, W_{i+1}]$; then, the weight inherited from ω_{jk} is $\omega_{jk} \cdot ((W_{i+1} - W_i)/(W_{jk+1} - W_{jk}))$. Following this strategy, WSHP traverses all contributors to derive all ω_i:

$$\omega_i = \sum_{j=1}^{M} \omega_{jk} \cdot \frac{W_{i+1} - W_i}{W_{jk+1} - W_{jk}}, \tag{19}$$

where $[W_i, W_{i+1}] \subset [W_{jk}, W_{jk+1}]$. Notice that $[W_i, W_{i+1}]$ will also belong to some $[W_{jk}, W_{jk+1}]$. Otherwise, $[W_i, W_{i+1}]$ will be further partitioned into subintervals.

Based on the weights, the data curator extracts corresponding sampling probabilities for different intervals. We consider a specific case where the incoming data are uniformly distributed over the whole range. Then, the sampling probabilities are determined by the following constraints. Firstly,

$$\frac{\sum_{i=1}^{K_0 - 1} P_i}{K_0 - 1} = P_0, \tag{20}$$

where P_0 is the overall ratio of collected bits. Secondly, for an arbitrary pair of P_i and P_j,

$$\frac{P_i}{P_j} = \frac{\omega_i\left(((W_{i+1} - W_i)/(D_U - D_L))\cdot N\right)^2 + \omega_i \alpha}{\omega_j\left(\left((W_{j+1} - W_j)/(D_U - D_L)\right)\cdot N\right)^2 + \omega_j \alpha}, \tag{21}$$

where $\alpha = (1/2)N^2 f$.

Finally, WSHP follows the same strategies with BSHP to iteratively sample bits from contributors based on the corresponding P_i and derives the results for different consumers.

4.3.1. Analysis. The objective of WSHP is to derive improved utilities for data consumers with heterogeneous concerns. The following theorem indicates that the WSHP algorithm can maximize the overall utility for all requestors.

Theorem 8. *With fixed privacy budgets and bandwidths, WSHP can achieve optimal utilities for data consumers when the data values uniformly are distributed in the whole range.*

Proof. As the bandwidths and privacy budgets are fixed in this case, WSHP adjusts the sampling ratios to balance the accuracy among different intervals. Meanwhile, the general variance is applied to measure the accuracy as WSHP provides an unbiased estimation. The general variance is

$$\mathrm{Var}(\mathcal{Q}) = \sum_{i=1}^{M} \mathrm{Var}(Q_i) = \sum_{i=1}^{M} \sum_{j=1}^{K_i - 1} \omega_{ij} \mathrm{Var}\left(R_{ij}\right). \tag{22}$$

Then, according to the correlations between ω_{ij} and ω_k and the relationships between intervals in the integrated partition and histograms, we have

$$\mathrm{Var}(\mathcal{Q}) = \sum_{j=1}^{K_0 - 1} \omega_j \mathrm{Var}\left(R_j\right). \tag{23}$$

Now we combine the analysis in equation (17) and derive

$$\mathrm{Var}\left(R_j\right) = \frac{\left(R_j^0\right)^2 + \alpha}{\beta P_j} + \gamma R_j^0 + \delta, \tag{24}$$

where α, β, γ, and δ are all constant, and $\alpha = (1/2)N^2 f$.

The results in equation (24) could be merged into equation (23),

$$\mathrm{Var}(\mathcal{Q}) = \sum_{j=1}^{K_0 - 1} \omega_j \left(\frac{\left(R_j^0\right)^2 + \alpha}{\beta P_j} + \gamma R_j^0 + \delta \right). \tag{25}$$

Minimizing the variance $\mathrm{Var}(\mathcal{Q})$ requests the knowledge on R_j^0s, which is obviously unavailable for data curators. Instead, we assume that the underlying data values are uniformly distributed in the range. Then, R_j^0 can be approximated by $((W_{j+1} - W_j)/(D_U - D_L)) \cdot N$. Therefore, the variance Var $(\mathcal{Q})$ is determined by $\sum_{j=1}^{K_0 - 1} (\omega_j (((W_{j+1} - W_j)/(D_U - D_L)) \cdot N)^2 + \omega_j \alpha)/P_j$. It is obvious that equation (21) can minimize $\mathrm{Var}(\mathcal{Q})$ in this circumstance, which is given in Theorem 8.

5. Discussion

This section covers the situations where data consumers may post their queries asynchronously, and the data curator has to acquire the data once and respond to continuously emerged queries.

The basic settings are similar to the previous cases, where the privacy budgets ε_0 and the bandwidth budgets B_0 are both fixed. In this case, the data curator could do the following:

(1) Devote all resources to extract one single histogram for all queries

(2) Partition the resource into multiple histograms and combine them for multiple queries

We will show that the first strategy is actually preferred even if it is straightforward. Initially, the derived results should try to provide an unbiased estimation for forthcoming queries. However, this is usually infeasible due to the diverse partition of intervals in histograms. Then, an alternative objective is to minimize the difference between the ground truth and the estimated result. In this worst case, the distance could be all data values falling in two consecutive intervals in the published histogram. Therefore, minimizing this distance leads to the identical most fine-grained partition on intervals, which implies the adoption of all resources.

On the other hand, we can also achieve the same conclusion by considering the use of privacy budgets. Intuitively, partitioning the budgets into multiple folds will not reduce the overall variance, while extra bandwidths will be wasted for content uploading. Therefore, it is also preferred that the first strategy should be selected for the online querying model.

Our future study will investigate the design of methods toward online histogram publication. Maybe other advanced mechanisms besides the random response will be introduced for this case.

Table 1: Statistics for datasets.

	Total contributors	Max salary	Min salary
Baltimore	13,683	250,000	1,800
New York	138,715	297,625	1
San Francisco	291,825	515,102	0

6. Evaluation

In this section, we adopt the salary data collected for normal citizens in the United States [33] to verify the performance of the sampling-based methods. New York City, San Francisco, and Baltimore are selected for our evaluation. Table 1 shows the overview of the datasets. We assume that data consumers request for the histogram of incoming levels with heterogeneous granularity. The data contributors will publish their data to the consumers, and the privacy concerns and bandwidth consumption should be treated. The data curator will coordinate the trading between two parties by generating the data plan and the final results.

The performance of the proposed algorithm is compared with a baseline method. In this method, the data contributors respond to each consumer separately. To thwart the collusion among consumers, the baseline algorithm requests the consumers to share the privacy budgets among multiple responses; e.g., assume the total privacy budgets to be ε_0; then, a contributor will apply ε_0/K budget to each of K queries. Each algorithm has been executed 20 times to mitigate the randomness. Finally, the mean square errors (MSE for short) are applied as the metric.

6.1. Basic Performance. This part studies both the numerical values and the overall performance. There are three

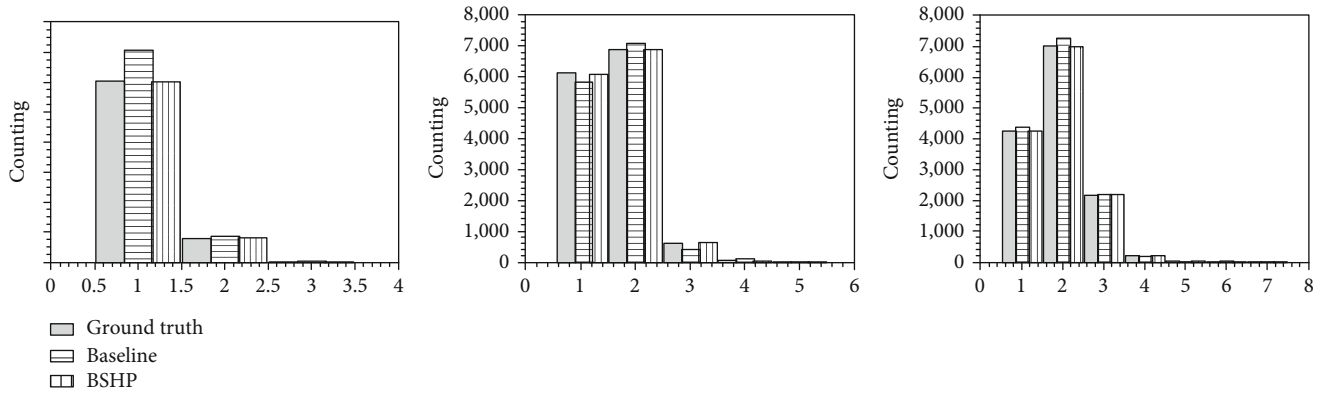


Figure 1: Multigranularity histograms for Baltimore.

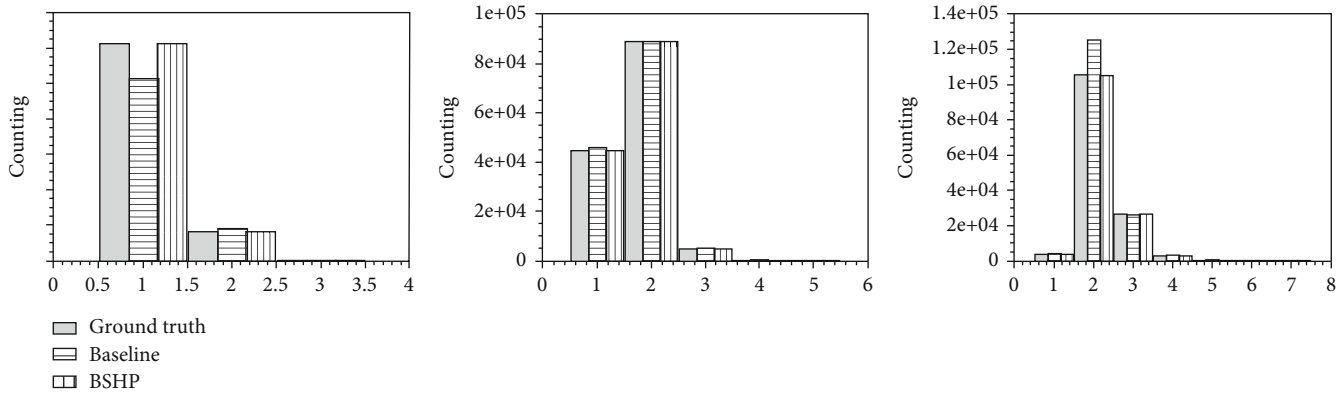


Figure 2: Multigranularity histograms for New York City.

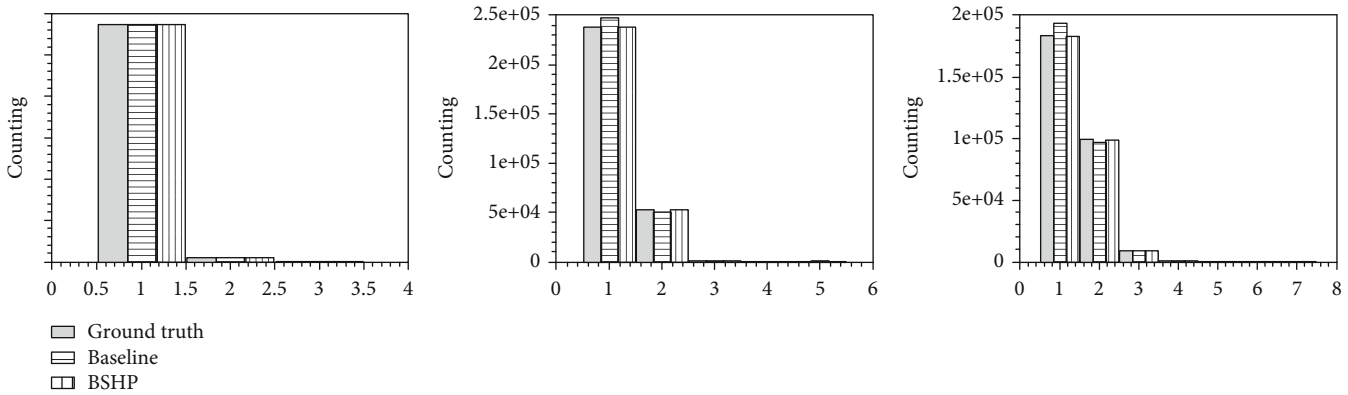


Figure 3: Multigranularity histograms for San Francisco.

consumers in the system, requesting 3-fold, 5-fold, and 7-fold histograms, respectively. They share the total budgets with $\varepsilon = 15$, where the baseline algorithm partitions the budgets among all three consumers. Meanwhile, our sampling-based algorithms apply all budgets for one common query. The sampling probability is 0.8.

The results are given in Figures 1–3. As we see, the proposed algorithms provide better utilities. They outperform the baseline method and achieve more accurate shapes for histograms, even though only part of the bits is collected under sampling. The difference is actually very significant when considering there are many data values belonging to some intervals to reduce the influence of randomness. It indicates that the proposed method can achieve good utilities by reduced bandwidth consumption.

This part also studies the overall performance under different budgets. In this group, the privacy budgets vary from 3 to 18. Two sampling-based algorithms are evaluated, with sampling probabilities set as 0.8 and 0.4.

According to the results in Figure 4, the proposed algorithms can reduce the MSE for histograms. The improvement is more significant when the privacy budget is

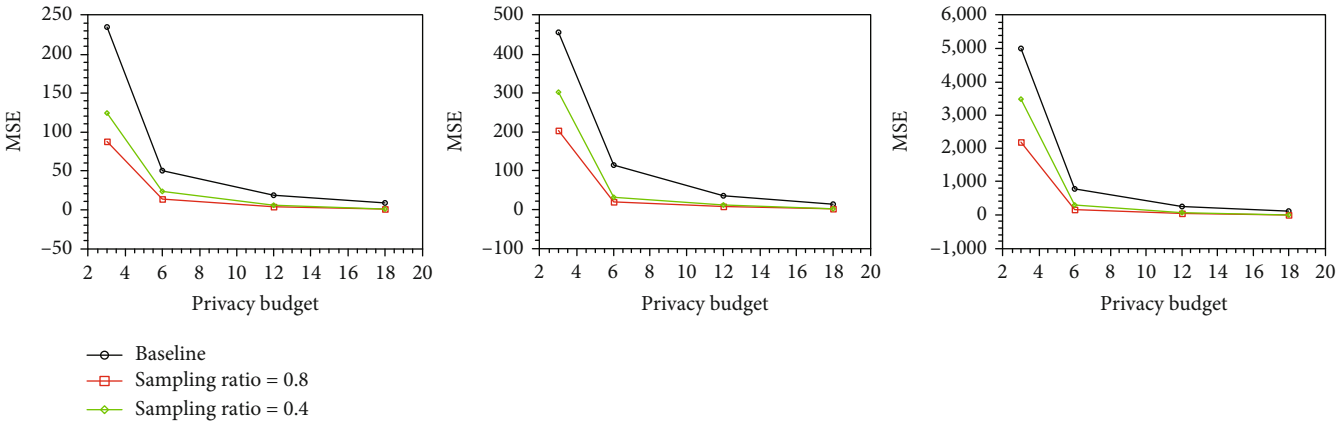


Figure 4: Mean square errors for histograms with various privacy budgets.

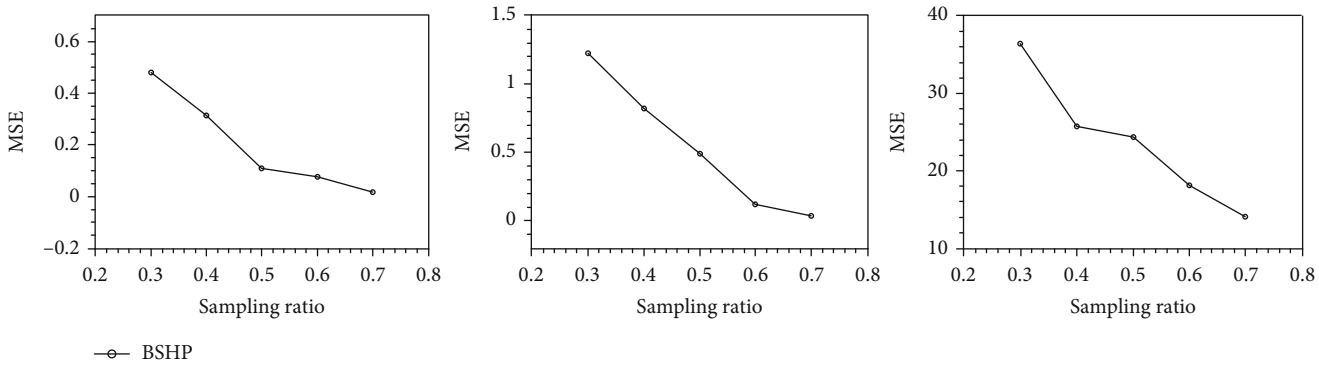


Figure 5: Mean square errors for histograms with various sampling ratios.

relatively large. The reason is that the saving on the privacy budget can overwhelm the effect of sampling so as to maintain a good utility. We also observe that a higher sampling ratio can improve the general performance. This is intuitively rational, as more samples could decrease the impacts of randomness.

6.2. Heterogeneous Sampling Ratios. Finally, we study the impact of various sampling ratios on histogram publication. In this group, the privacy budget is 15, and the data consumers still request 3-fold, 5-fold, and 7-fold histograms. The sampling ratios increase from 0.3 to 0.7 with the incremental step as 0.1. The results are depicted in Figure 5.

The general performance is improved for all three datasets according to the results, where the MSE value is reduced by at least 50% (San Francisco). However, the performance actually stays on the same scale for each dataset. This observation implies that increasing the sampling ratio (i.e., the bandwidth) will not always improve the data utility. In this case, the privacy budget will become the bottleneck for highly effective data publication. However, it also indicates that the bandwidth could be saved while the total utilities will not be reduced by too large.

Generally, both proposed algorithms can effectively and efficiently improve the performance of histogram publication.

7. Conclusion

To jointly preserve sensitive information and improve efficiency during data collection has long been considered a challenging task for data processing. The emergence of local differential privacy sheds light on this task. However, existing works fail to combine the sampling strategy with the mechanisms designed for LDP. Therefore, this work proposes a novel framework for privacy-preserved histogram publication in distributed manners. It first investigates a novel plan for data collection over numerical values and then designs two sampling-based algorithms for data encoding and decoding. These algorithms apply bit-level sampling to balance the cost among data contributors and can help consumers adjust their devotion on different intervals of the histogram. Extensive analysis is proposed, including unbiased results, privacy preservation, and optimization in allocating the bandwidth resources. Finally, we conduct an evaluation on one real-world dataset to show the superiority of proposed algorithms.

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 conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. U19A2059 and 61802050), the National Key R&D Program of China (No. 2018YFC0807500), and the Ministry of Science and Technology of Sichuan Province Program (Nos. 2018GZDZX0048 and 20ZDYF0343).

References

[1] X. Wang, L. T. Yang, Y. Wang, L. Ren, and M. J. Deen, "ADTT: a highly efficient distributed tensor-train decomposition method for IIoT big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1573–1582, 2020.

[2] C. Lim, K.-J. Kim, and P. P. Maglio, "Smart cities with big data: reference models, challenges, and considerations," *Cities*, vol. 82, pp. 86–99, 2018.

[3] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[4] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, and M. J. Deen, "A tensor-based big-data-driven routing recommendation approach for heterogeneous networks," *IEEE Network*, vol. 33, no. 1, pp. 64–69, 2019.

[5] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, 2013.

[6] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, pp. 127–135, Portland, OR, USA, June 2015.

[7] M. Bun, J. Nelson, and U. Stemmer, "Heavy hitters and the structure of local privacy," in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 435–447, San Francisco, California, USA, 2018.

[8] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, 2018.

[9] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[10] X. Zheng, Z. Cai, J. Li, and H. Gao, "Locationprivacy-aware review publication mechanism for local business service systems," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, Georgia, USA, 2017.

[11] X. Wang, L. T. Yang, L. Song, H. Wang, L. Ren, and M. J. Deen, "A tensor-based multiattributes visual feature recognition method for industrial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2231–2241, 2020.

[12] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, Scottsdale, Arizona, USA, 2014.

[13] K. Micinski, P. Phelps, and J. S. Foster, "An empirical study of location truncation on android," *Weather*, vol. 2, p. 21, 2013.

[14] R. Chow and P. Golle, "Faking contextual data for fun, profit, and privacy," in *Proceedings of the 8th ACM workshop on Privacy in the electronic society - WPES '09*, pp. 105–108, Chicago, Illinois, USA, 2009.

[15] J. Wang, Z. Cai, and J. Yu, "Achieving personalized *k*-anonymity-based content privacy for autonomous vehicles in cps," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4242–4251, 2020.

[16] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 754–762, Toronto, ON, Canada, April 2014.

[17] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1021–1032, 2010.

[18] G. Acs, C. Castelluccia, and R. Chen, "Differentially private histogram publishing through lossy compression," in *2012 IEEE 12th International Conference on Data Mining*, pp. 1–10, Brussels, Belgium, December 2012.

[19] Y.-H. Kuo, C. C. Chiu, D. Kifer, M. Hay, and A. Machanavajjhala, "Differentially private hierarchical count-of-counts histograms," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1509–1521, 2018.

[20] X. Zhang, R. Chen, J. Xu, X. Meng, and Y. Xie, "Towards accurate histogram publication under differential privacy," in *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 587–595, Philadelphia, Pennsylvania, USA, April 2014.

[21] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[22] X. Zheng, G. Luo, and Z. Cai, "A fair mechanism for private data publication in online social networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 880–891, 2020.

[23] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[24] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 425–438, Dallas, Texas, USA, October 2017.

[25] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, July 2019.

[26] S. Wang, L. Huang, P. Wang, H. Deng, H. Xu, and W. Yang, "Private weighted histogram aggregation in crowdsourcing," *International Conference on Wireless Algorithms, Systems, and Applications*, 2016, pp. 250–261, Springer, 2016.

[27] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. of the 26th USENIX Security Symposium*, pp. 729–745, Vancouver, BC, Canada, 2017.

[28] J. Soria-Comas and J. Domingo-Ferrer, "Optimal data-independent noise for differential privacy," *Information Sciences*, vol. 250, pp. 200–214, 2013.

[29] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath, "The staircase mechanism in differential privacy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1176–1184, 2015.

[30] N. Wang, X. Xiao, Y. Yang et al., "Collecting and analyzing multidimensional data with local differential privacy," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 638–649, Macao, China, April 2019.

[31] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[32] N. Li, W. Qardaji, and D. Su, "On sampling, anonymization, and differential privacy or, kanonymization meets differential privacy," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 32-33, Singapore, 2012.

[33] "Data.world," https://data.world/datasets/salary.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6632671, 9 pages
https://doi.org/10.1155/2021/6632671

Research Article

Mining Network Traffic with the k-Means Clustering Algorithm for Stepping-Stone Intrusion Detection

Lixin Wang,[1] **Jianhua Yang**,[1] **Xiaohua Xu**,[2] **and Peng-Jun Wan**[3]

[1]*TSYS School of Computer Science, Columbus State University, Columbus, Georgia, USA*
[2]*College of Computing and Software Engineering, Kennesaw State University, Kennesaw, Georgia, USA*
[3]*Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA*

Correspondence should be addressed to Lixin Wang; wang_lixin@columbusstate.edu

Received 15 December 2020; Revised 19 January 2021; Accepted 6 February 2021; Published 3 March 2021

Academic Editor: Wenzhong Li

Intruders on the Internet usually launch network attacks through compromised hosts, called stepping stones, in order to reduce the chance of being detected. With stepping-stone intrusions, an attacker uses tools such as SSH to log in several compromised hosts remotely and create an interactive connection chain and then sends attacking packets to a target system. An effective method to detect such an intrusion is to estimate the length of a connection chain. In this paper, we develop an efficient algorithm to detect stepping-stone intrusion by mining network traffic using the k-means clustering. Existing approaches for connection-chain-based stepping-stone intrusion detection either are not effective or require a large number of TCP packets to be captured and processed and, thus, are not efficient. Our proposed detection algorithm can accurately determine the length of a connection chain without requiring a large number of TCP packets being captured and processed, so it is more efficient. Our proposed detection algorithm is also easier to implement than all existing approaches for stepping-stone intrusion detection. The effectiveness, correctness, and efficiency of our proposed detection algorithm are verified through well-designed network experiments.

1. Introduction

This paper is an extension of our work originally presented at the 39th IEEE International Performance Computing and Communications Conference (IEEE IPCCC 2020) [1]. Many attackers send attacking packets to remote target systems through compromised machines, for the purpose of decreasing the chance of being discovered [2–11]. The compromised machines employed by the attackers are referred to as stepping stones. In a stepping-stone intrusion (SSI), an intruder uses a chain of compromised machines on the Internet as relay hosts and remotely logs in these machines by using software tools such as SSH, rlogin, or telnet. The attacker sits in front of his local host and types attacking commands that are relayed via the stepping-stone hosts in the connection chain until the attacking packets arrive the remote target system that is under attack.

Since every TCP connection between a source node and a destination node is independent of other connections even though the connections might be relayed, accessing a remote host via several relayed TCP connections makes it very difficult to determine the attacker's actual geographical location. Because the TCP protocol has such a property, the final target machine could only see the packets from the last connection of the chain. Therefore, it is extremely hard for a target host to learn any information about the actual location of the intruder.

A benefit of launching attacks using stepping stones is that attackers could be hidden behind a long interactive connection. If a SSI could be detected within the active period of attacking, then the session could be cut off and the target system could be protected. Although some researchers worked on the back-tracing of SSI and studied the upstream detection, most researchers focused on downstream SSI detection.

Intruders using SSI could build a connection chain given in Figure 1 using software tools such as SSH to launch their attacks. In Figure 1, we assume that Host 0 is used by the

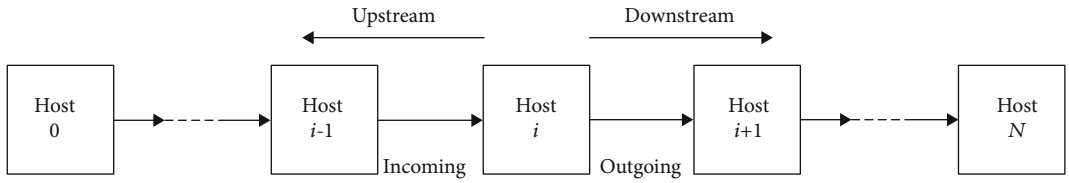


Figure 1: A sample connection chain.

attacker to launch an attack against the target Host N via compromised hosts Host 1, Host 2,…, Host $i-1$, Host i, Host $i+1$,…, and Host $N-1$. SSI detection may occur in any of these stepping stones. The detection program is assumed to reside in Host i which is referred to as a (detecting) sensor. SSI detection is to determine if the detecting sensor Host i is used as a stepping stone. The connection from Host $i-1$ to Host i is called an incoming connection to Host i, and the connection from Host i to Host $i+1$ is called an outgoing connection from Host i. If there is at least one relayed pair between all the incoming connections and all the outgoing connections, then it is highly suspicious that Host i is used as a stepping stone for intrusion.

One type of detection methods for SSI is to compare all the outgoing connections with all the incoming connections of the same host to see if there exists a relayed pair. This type of methods is referred to as *host-based* SSI detection. Quite a few approaches that can be applied to encrypted connections have been proposed since the year 1995 including the deviation-based [12] and time-based approaches [3, 4, 6, 13–18] for SSI detection. This type of methods only focuses a single host and requires only the outgoing packets leaving the host and incoming packets to the host. The main disadvantage of this type of approaches is that it usually introduces high false-positive errors as some legal applications actually use stepping stones to access remote servers. For example, when a client browser requests some resources from a Web server, the Web server may need to access a remote application server that also may need to access a remote database server. Moreover, this type of detection approaches is also vulnerable to the intruder's session manipulation that can be done by using hacking techniques such as time-jittering and/or chaff permutation.

The other type of detection methods to address the problems with the host-based methods is to estimate the number of connections from the intruder's host to the target host (as shown in Figure 1), which is called *the length of a connection chain*. If there exist three or more connections in a connection chain, it means that the user attempts to gain access to a remote target host via three or most machines. It is well-known that the more computers involved in an interactive connection to gain access to a remote server, the slower the traffic of data communication. The threshold number "three" was discovered because most legal applications seldom used three or more stepping stones to access a remote server. This type of detection methods is referred to as the *connection-chain-based or network-based* SSI detection.

However, the connection-chain-based detection methods could produce a false negative error. Let us first introduce two concepts of *upstream and downstream detections*. Estimating the length of the connection chain from the sensor (Host i) to attacker's host (Host 0) as shown in Figure 1 is referred to as *upstream detection*. Similarly, estimating the length of the connection chain form the sensor (Host i) to the target host (Host N) is referred to as *downstream detection*. The length of the entire connection chain from attacker machine to the target machine equals the length of the downstream chain plus that of the upstream one. Unfortunately, upstream detection is extremely challenging and has been a long-standing open problem. Therefore, it is very hard to estimate the length of the entire chain since we can only estimate the length of the downstream connection chain. When the detecting sensor is close to the target machine, the length of the downstream connection chain is trivial, and the upstream connection would dominate the length of the entire chain. In such a case, all malicious intrusions will escape detection due to the false negative errors.

If the downstream connection length is at least two (it means that at least two stepping-stone hosts are present between the sensor computer to the target host) plus the upstream connection length; this makes the length of the whole connection chain be three or more. Therefore, when the downstream connection length is at least two, we conclude that it is highly suspicious that the session is initiated by an attacker. Most existing detection algorithms by estimating the length of a connection chain developed by far only takes the downstream connection into consideration. In this work, the length of a connection chain always means the length of its downstream connection.

Let us give a literature review on network-based detection methods for SSI. The first detection algorithm using a network-based approach was proposed by Yung [19] in 2002. Its basic idea is to estimate the length of a connection chain by computing the RTT of a Send packet and matching it with an ACK packet sent from an adjacent node. Although the false positive error incurred by the approach proposed in [19] is reduced a little bit, this detection approach also suffers from producing false negative errors occasionally, because the method developed in [19] used the ACK packets instead of the actual echo packets. Based on the connection chain set up in [19], the actual echo packets are not available to capture.

Yang et al. [20] developed the step-function detection approach to estimate a connection chain length in a LAN. Compared to the detection algorithms proposed in [19], the step-function approach reduced both the false positive error and the false negative error. In [20], the first connection chain created contains only one connection. Then, the connection chain is extended to two connections, three connections, and so on.

The key idea of the step-function method is to obtain the RTTs by matching a Send packet with its corresponding Echo packet which is available based on the way the connections are set up. However, the approach proposed in [20] only worked when the data communications are limited within a local area network. In the context of the Internet, the same set of authors in [21] proposed the conservative and greedy packet matching algorithm for SSI detection. The weakness of such a conservative algorithm is that it can only match very few packets, which makes the detection method ineffective.

The best known connection-chain-based detection method for SSI is the clustering and partitioning data mining algorithm proposed by Yang and Huang [22]. This paper developed an algorithm for SSI detection by using a clustering and partitioning data mining approach to compute the RTTs of the packets captured from a connection chain. All of the previously known approaches of matching Send and Echo packets only compare one Echo packet with a Send packet at a time. The method proposed in [22] looked through all the possible packets to produce TCP packet matches and made the matching accurate. Using the method in [22], all the Send and Echo packets are captured from a connection chain in a time interval, and then the timestamp differences between a Send packet and those Echo packets received after that Send packet are all calculated. The method proposed in [22] made sure that the correct RTT of each Send packet is among these timestamp differences. The maximum–minimum distance clustering algorithm (MMD) was used to computer the real RTTs. The number of connections in the chain is determined based on the number of clusters generated by the MMD data mining algorithm. The results obtained from the well-designed experiments in [22] indicated that this approach can more accurately estimate the connection chain length than all of the prior detection algorithms for SSI and reduce both the false positive and false negative errors.

A weakness of the MMD clustering and partitioning approach developed in [22] is that a huge number of packets have to be captured and processed. Therefore, this algorithm is not efficient. This paper addresses these issues by introducing a novel algorithm to detect SSI via mining network traffic using the k-means clustering algorithm [23–26]. The k-means clustering algorithm is a data mining approach used to cluster observations into groups of related observations without any prior knowledge of those relationships. It is well-known that the round-trip times will cluster around several levels. As long as most of the outlier values can be removed from the real RTTs in the input file, our k-means clustering detection algorithm proposed in this work can accurately determine the length of a connection chain. Also, our proposed detection algorithm does not require a large number of packets to be captured and processed. Thus, our proposed method for SSI detection is more efficient than MMD clustering and the partitioning approach developed in [22]. The effectiveness of our innovative approach for SSI detection is verified through well-designed experiments by appropriately setting up the connection chains. The experimental result showed that our proposed algorithm can estimate the length of a connection more accurate than the known detection methods presented in the literature.

Table 1: All notations used in this paper.

X	A random variable
μ	Mean of a random variable
σ	Standard derivation
k	The number of clusters
c_i	The center of the i-th clusters, $i = 1, 2, \cdots, k$
C	The set of k centers $\{c_1, c_2, \cdots, c_k\}$
Dataset-j	A dataset of RTT values; $j = 1$, 2, or 3
σ_{curr}	Standard derivation based on the current cluster
σ_{new}	Standard derivation based on the updated cluster

It is worth mentioning some other work related to SSI detection and/or network security. In order to help researchers to understand how a session is chaffed and develop new tools for detecting SSI as well as resisting intruders' chaff manipulation, [27] developed a C# program to inject meaningless packets into a TCP/IP session. Reference [28] presented an adaptive network intrusion detection method using fuzzy rough set-based feature selection and pattern learning. This paper also introduced a greedy approach of global optimal Gaussian mixture model clustering method in order to extract the intrinsic structure of network instances to achieve highly discernable and stable normal and intrusion pattern libraries for the subsequent network intrusion detection. Many methods have been proposed to detect stepping stones and resist evasive behavior, but so far, no benchmark dataset exists to provide a fair comparison of detection rates. Reference [29] developed a comprehensive framework to simulate realistic stepping-stone behavior that includes effective evasion tools and release a large dataset. The detection rates of eight state-of-the-art methods are evaluated by using this framework. Reference [30] studied query-efficient adversarial attacks against automatic speech recognition systems. This paper presented a novel and effective attack on automatic speech recognition systems. Compared with prior known methods in the literatures, the approach proposed in [30] only needs limited access to the output probabilities of neural networks and achieves extremely high efficiency and success rates. Data privacy relates to how data should be handled based on its relative importance and is closely relevant to network security. Reference [31] developed an approach for uploading data in smart cyberphysical systems, in which both energy conservation and privacy preservation are taken into consideration. Reference [32] proposed a privacy-preserved data sharing structure for industrial Internet of Things, in which several competing clients could exist in distinct stages of the system. Reference [33] developed a mechanism for trading range counting results. Under differential privacy, this mechanism used a sampling method to produce rough counting results which are theoretically verified to achieve unbiasedness, bounded variance, and privacy guarantee.

All the notations used in this paper are listed in Table 1 for easy referencing.

The remaining of this paper is organized as follows. In Section 2, we provide some preliminaries needed for the detection algorithm design. In Section 3, we present our algorithm for SSI detection by mining network traffic using k -means clustering. Performance analysis of our proposed detection algorithm is given in Section 4. Finally, we summarize our paper and discuss some future research directions in Section 5.

2. Preliminaries

In this section, we introduce some basic concepts in computer networks that are required to design our detection algorithm for SSI and the rationale of using packets' RTTs to estimate the length of a connection chain.

2.1. Definitions of Send/Echo Packets. Let us use Figure 1 to define Send and Echo packets. Host i is the detecting sensor. In the incoming connection of Host i, a Send packet is defined as a TCP packet received at Host i and sent from Host $i-1$, with the TCP.Flag.PSH it set; an Echo packet is defined as a TCP packet received at Host $i-1$ and sent from Host i, with the TCP.Flag.PSH bit set. In the outgoing connection from Host i, a Send packet is defined as a TCP packet received at Host $i+1$ and sent from Host i, with the TCP.Flag.PSH bit set; an Echo packet is defined as a TCP packet received at Host i and sent from Host $i+1$, with the TCP.Flag.PSH bit set.

Now let us use an example to explain which Send packet and Echo packet are *a matched pair*. When a user types a command on a command line in a Linux system, such as "ls," it might be sent to the server in one or two packets. Suppose that the command "ls" is sent to the remote server in two separate Send packets: "l" and "s." When "l" is typed on the user's command line, the packet will be sent to the server side. Once this Send packet is echoed, an Echo packet sent back to the user's host, letter "l" will be shown on the terminal of the user's host. Such Send and Echo packets are called *a matched pair*. For the other letter "s," a matched pair can be similarly obtained: a Send "s" and an Echo "s." Using the timestamps of a matched pair of Send and Echo packets, their packet RTT can be easily computed. The length of the connection time of an interactive TCP session is represented by the RTT of a matched pair. The RTT computed from the matched pair of the Send and Echo packets of "l" is different from the RTT computed from the matched pair of the Send and Echo packets of "s"; these two numbers are very close to each other because these two RTTs stand for the length of the same connection in different time periods. A Send packet could be echoed by one or Echo packets. Also, an Echo packet could echo one or more Send packets.

2.2. The Distribution of Packets' RTTs for a Connection Chain. A packet RTT of a TCP connection is the sum of four time delays including the processing delay, queuing delay, transmission delay, and propagation delay of the underlying connection [34]. For connection-chain-based SSI detection, the length of a connection chain is estimated by using the packet RTTs. The RTTs obtained from matched Send and Echo pairs can be used to represent the network traffic. Yang et al. [35] showed that the length of a connection chain is equal to the number of clusters that are produced by using the RTTs obtained from the connection chain.

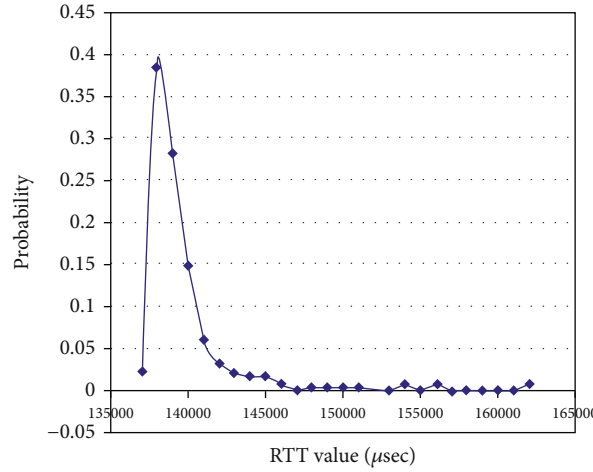


Figure 2: The distribution of packets' RTTs for a connection chain.

Paxson and Floyd [4] found the fact that packet RTTs obtained from a connection chain obey Poisson distribution. This can be used to match TCP packets and further estimate the number of connections in an extended connection chain. Figure 2 shows a typical experiment from which packet RTTs obey Poisson distribution, where the y-axis represents the probability of the occurrence of each RTT, and the x-axis represents the values of the RTTs with unit of microsecond. The values of the RTTs shown in Figure 2 were from the packets collected from a connection chain composed of four connections. With this experiment, most values of the RTTs are close to the mean $\mu = 138{,}500$ (microsecond) of all the RTTs, with at least 95% of the RTT values in the range between 137,000 (microsecond) and 141,000 (microsecond) (refer to Figure 2).

Assume that X is a random variable with mean μ and standard deviation σ. If X follows the Poisson distribution, then we have

$$|X - \mu| \leq 2\sigma. \tag{1}$$

This inequality tells us that most values that X takes should be close to its mean μ. The difference between X and its mean μ is upper bounded within 2σ. That is, most RTT values of a connection chain with fixed length must be around its mean within a circle with radius 2σ. Therefore, the RTTs corresponding to a connection chain with fixed length belong to the same cluster with the mean μ as the cluster center. Consequently, the RTT values of a connection chain with distinct lengths belong to different clusters. Thus, if we can design an algorithm to obtain the number of such clusters, then we could easily obtain the matched packets and the length of a connection chain.

It is well-known that many applications such as Web applications use a host as a stepping stone to access a remote server. But for most legitimate applications, it is very seldom

to use three or more hosts as stepping stones to access a remote server. The more hosts that are used to access a remote server, the slower the data communication. If there is no intention to hide malicious activities, it is unnecessary to access a remote server via three or more stepping stones as it could make the remote accessing inefficient. Therefore, it is reasonable to assume that if a host uses three or more stepping stones to access a remote server, that host is highly suspicious as a malicious intruder.

3. Stepping-Stone Intrusion Detection Using k -Mean Clustering

In this section, we develop an algorithm for SSI detection by mining network traffic using k-means clustering. Initially, we create a connection chain with only one connection from the hacker's host to the first stepping-stone host that will be used as the sensor where a detection program such as TCPDump is installed. Then, we extended the connection chain to a chain containing two connections from the sensor host to the target host. Following that, the chain is extended to contain three connections and finally four connections. The length of a connection chain means that of the downstream chain from the sensor host to the target machine. For the connection chain of two connections, its TCP Send and Echo packets are captured, and then the RTTs of the chain are calculated. Since the packets' RTTs of a connection chain follow Poison distribution, most RTTs are around the mean of the RTTs of the connection chain. For the connection chain of three connections, most of its RTTs are similarly around the mean of the RTTs of this chain. The same is true for the connection chain of four connections.

Many data mining clustering algorithms were proposed (see [22] and there references therein). Among all these known data mining clustering approaches, the k-means clustering method is a simple unsupervised learning algorithm that gives a solution to the well-known clustering problem [23–26]. In this paper, the k-means clustering *algorithm* is used in the design of our detection algorithm for SSI. With the k-means clustering *algorithm*, k is the number of clusters that is fixed and predetermined.

The key idea is to find the appropriate k centers, one for each cluster. These k centers need to calculate in a cunning approach so that the overall standard derivation calculated based on these k centers is minimized. It is easy to see that we need to place the k centers as far away as possible from each other. References [23–26] gave detailed description for the k-means algorithm and explanation how this algorithm can be implemented. The k-means clustering algorithm is a perfect choice of data clustering in this paper due to the unique properties of the dataset computed from the time-stamp differences of Send and Echo packets. Such a method is one of the best approaches that can accurately estimate a connection chain length. Because most legal applications seldom employ three or more intermediate hosts to access a remote server, if a target system is accessed via three or more stepping stones, then it is most likely that the session is manipulated by a malicious hacker for SSI. Therefore, it is sufficient to only use the three values 2, 3, and 4 for the value of k, respectively.

Next, we describe the rationale of designing detection algorithms for SSI by using the k-means clustering, given three RTT datasets obtained from connection chains of length two, three, or four, respectively. But we do not know which dataset was collected from a connection chain of length two, three, or four. When $k = 2$, we run the k-mean clustering algorithm on these three RTT datasets. It is observed that the output generated by the dataset corresponding to the connection-chain length two would produce the smallest standard derivation. According to such an observation, for these three datasets, we may conclude that the dataset that produced the output with the smallest standard derivation was obtained from the connection chain of length two. We have a similar observation for $k = 3$. We execute the k-means clustering algorithm on these three datasets. It is observed that the output generated by the dataset corresponding to the connection-chain length three would produce the smallest standard derivation among the three datasets. According to such an observation, for these three datasets, we may conclude that the dataset that produced the output with the smallest standard derivation was obtained from the connection chain of length three. For $k = 4$, we also have a similar observation. As a result, by mining network packets using the k-means clustering method, we can estimate the length of a connection chain.

Suppose that $Y = \{y_1, y_2, \cdots, y_n\}$ is a set of n data points. Let $C = \{c_1, c_2, \cdots, c_k\}$ be the set of k centers that will be used by the k-mean clustering algorithm. We partition Y into k disjoint subsets $Y_1, Y_2, \cdots, Y_k$, where the data points in Y_j are associated with their nearest center c_j in C for each $1 \le j \le k$ by associating each point in Y to the nearest center in C. For each $1 \le j \le k$, let $Y_j = \{y_{j_1}, y_{j_2}, \cdots, y_{j_Y}\}$. The standard derivation σ of the given dataset Y according to the k-means clustering is defined to be the square root of its variance:

$$\sigma = \sqrt{\frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{|Y_j|}\left\|y_{ji} - c_j\right\|^2}. \tag{2}$$

Let dataset-1, dataset-2, and dataset-3 denote the three RTT datasets obtained from three connection chains of distinct length 2, 3, or 4, respectively. But we do not know which RTT dataset was obtained from which connection chain beforehand. Our proposed algorithm for SSI detection using the k-means clustering method can accurately determine which RTT dataset was obtained from the connection chain of length 2, 3, or 4, respectively. Our proposed detection algorithm for SSI is described in Algorithm 1.

Let us explain Algorithm 1 that is used to estimate the length of a connection chain. The value of the integer k is between 2 and 4.

At Step 1, the k-means clustering algorithm is called on dataset-1. Here is the procedure: initially, we randomly select k points in Y as the k centers for the clustering algorithm that executes the steps below:

Input: k, dataset-1, dataset-2, and dataset-3.
Output: the RTT dataset obtained from the connection chain of length k.
(1) Call the k-means clustering algorithm on dataset-1. Assume σ_1 represents the standard derivation outputted based on Equation (2) using the k clusters obtained at the end of the k-means clustering algorithm execution
(2) Call the k-means clustering algorithm on dataset-2. Assume σ_2 represents the standard derivation outputted based on Equation (2) using the k clusters obtained at the end of the k-means clustering algorithm execution
(3) Call the k-means clustering algorithm on dataset-3. Assume σ_3 represents the standard derivation outputted based on Equation (2) using the k clusters obtained at the end of the k-means clustering algorithm execution
(4) If $\sigma_1 = \min\ \{\sigma_1, \sigma_2, \sigma_3\}$, **return** dataset-1; /* if σ_1 is the smallest one among all three standard derivations, the dataset-1 is obtained from the connection chain of length k */
(5) If $\sigma_2 = \min\ \{\sigma_1, \sigma_2, \sigma_3\}$, **return** dataset-2; /* if σ_2 is the smallest one among all three standard derivations, the dataset-2 is obtained from the connection chain of length k */
(6) If $\sigma_3 = \min\ \{\sigma_1, \sigma_2, \sigma_3\}$, **return** dataset-3; /* if σ_3 is the smallest one among all three standard derivations, the dataset-3 is obtained from the connection chain of length k */

Algorithm 1: Algorithm to estimate the length of a connection chain

(i) To generate the k clusters, each point in Y is scanned and assigned to the cluster center whose distance from the cluster center is minimum of all these cluster centers

(ii) Calculate the standard derivation σ_{curr} using the current partition and cluster centers according to Equation (2)

(iii) For each cluster, recompute the new cluster center which is the average of the data points in the cluster

(iv) Use these new cluster centers to reproduce the k clusters following the same procedure described in step (i)

(v) Recompute the standard derivation σ_{new} using the updated partition and cluster centers according to Equation (2)

(vi) If $\sigma_{\text{new}} \geq \sigma_{\text{curr}}$ then Exit; otherwise, go back and repeat step (iii) above

When the k-means clustering algorithm exits, the standard derivation is minimized using the cluster centers and partition obtained at the last round of the algorithm. Thus, after the algorithm's last round completes, the standard derivation can no longer be smaller through changing the positions of the cluster centers. According to Equation (2), the value of σ_1 is the standard derivation obtained by using the k clusters calculated at the last round of the k -means clustering.

Similarly, at Steps 2 and 3 of Algorithm 1, the k-means clustering algorithm is called on dataset-2 and dataset-3, respectively. Then, σ_2 and σ_3 are in turn computed. If σ_1 is minimum among the three values σ_1, σ_2, and σ_3, then the dataset dataset-1 is the one that was obtained from the connection chain of length k. The similar conclusions are also true for the two datasets dataset-2 and dataset-3.

Finally, Algorithm 1 will be executed for the values of $k = 2, 3,$ and 4. Hence, Algorithm 1 can accurately determine the length of each of these three connection chains from which the three RTT datasets were collected.

4. Performance Analysis of the Proposed Detection Algorithm

In this section, we analyze the performance of our proposed Algorithm 1 above for SSI detection by mining network traffic using the k-means clustering approach. The window size we used to generate the RTT datasets in the experiments is three for all the captured packets. That is, for each Echo packet, we compute the timestamp differences between this Echo packet and up to three previous Send packets. We will verify the effectiveness and correctness of our proposed detection algorithm for SSI through well-designed network experiments. In total, we collected 20 collections of datasets, each of which contains three datasets obtained from connection chains of length 2, 3, or 4, respectively. Assume that we do not know the length of the connection chain where each of these datasets was collected from. Our proposed detection algorithm can accurately determine which dataset was collected from a connection chain of length 2, which dataset from a connection chain of length 3, and which dataset from a connection chain of length 4.

In our experiment, we initially created a connection chain with only one connection from the intruder's host to the first stepping-stone host S1 that will be used as the detecting sensor where a detection program such as TCPDump is installed. Then, the connection chain is extended to the host S2 and then to the host S3. Now the length of the downstream chain (from the sensor S1 to the host S3) is two. After that, the connection chain is extended to contain three connections and, finally, up to four connections in the downstream connection from the sensor host S1 to the victim host. See Figure 3 for the connection chain of length four.

For the connection chain of two connections, we monitored and captured its TCP Send and Echo packets and then computed the RTTs for the connection chain. The RTT datasets were generated by using a window of size three. We found that most of its RTTs are actually around the mean of these RTTs of the connection chain. Theoretically, this is true according to the distribution of the packets' RTTs of a connection chain we discussed in Section 2. Then, we

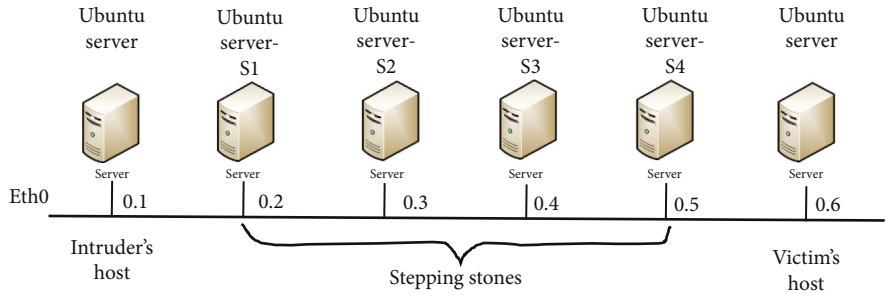


Figure 3: A connection chain of length four (from the sensor host S1 to the victim's host).

Table 2: The standard derivations outputted by the 2-mean clustering algorithm on the three RTT datasets collected from connection chains of lengths 2, 3, and 4, respectively.

RTT dataset	Chain length	Standard derivation
Dataset-1	2	678238.35
Dataset-2	3	760574.99
Dataset-3	4	758813.90

Table 3: The standard derivations outputted by the 3-mean clustering algorithm on the three RTT datasets collected from connection chains of lengths 2, 3, and 4, respectively.

RTT dataset	Chain length	Standard derivation
Dataset-1	2	68994.78
Dataset-2	3	5377.87
Dataset-3	4	6621.19

perform the same procedures for the connection chains of three and four connections, respectively. For the connection chain of length three, most of its RTTs are indeed around the mean of the RTTs of the connection chain. The same is true for the connection chain with length four. Let dataset-1, dataset-2, and dataset-3, respectively, represent the datasets of packets RTTs generated from the captured TCP Send and Echo packets collected from the connection chains of lengths 2, 3, and 4, using a window of size 3.

Then, we run the 2-means clustering algorithm (i.e., all items in the dataset are divided into 2 clusters) on the datasets dataset-1, dataset-2, and dataset-3. For each input dataset, this 2-means clustering algorithm outputs the standard derivation calculated using the 2 clusters obtained at the last iteration of the 2-means clustering based on Equation (2). The standard derivations outputted by this clustering algorithm are listed in Table 2. Based on this table, the RTT dataset we collected from the connection chain of length 2 achieves the smallest standard derivation in this case.

Similarly, we run the 3-means clustering algorithm (i.e., all items in the dataset are divided into 3 clusters) on the datasets dataset-1, dataset-2, and dataset-3. The standard derivations outputted by this clustering algorithm are listed in Table 3. Based on this table, the RTT dataset we collected from the connection chain of length 3 achieves the smallest standard derivation in this case.

Table 4: The standard derivations outputted by the 4-mean clustering algorithm on the three RTT datasets collected from connection chains of lengths 2, 3, and 4, respectively.

RTT dataset	Chain length	Standard derivation
Dataset-1	2	7542.40
Dataset-2	3	1870.36
Dataset-3	4	1733.15

Finally, we run the 4-means clustering algorithm (i.e., all items in the dataset are divided into 4 clusters) on the datasets dataset-1, dataset-2, and dataset-3. The standard derivations outputted by this clustering algorithm are listed in Table 4. Based on this table, the RTT dataset we collected from the connection chain of length 4 achieves the smallest standard derivation among all three datasets.

Therefore, we can accurately determine the length of the connection chain where each of the three RTT datasets was collected from by using our proposed detection algorithm based on the k-means clustering.

5. Conclusion and Future Research Directions

We proposed in this paper an efficient detection algorithm for SSI by using the k-means clustering algorithm to estimate the length of a (downstream) connection chain. Our proposed detection algorithm using the k-means clustering can accurately determine the connection chain length. Also, the proposed algorithm does not require capturing and processing a large number of TCP packets. Therefore, our proposed detection algorithm for SSI is efficient. Due to the use of the k-means clustering approach, our proposed algorithm is easy to implement as well. Through well-designed network experiments by setting up three connection chains, the effectiveness and correctness of our proposed detection algorithm for SSI are verified. For future research work, one can appropriately modify some steps of the k-means clustering algorithm so that the detection algorithm for SSI can significantly reduce both the false positive and false negative errors compared to all existing approaches. Also, the detection methods based on the k-means clustering require that there should be as less outlier values of round-trip times as possible. Therefore, additional algorithms are needed to remove these outlier values of round-trip times from the input file of the detection algorithm.

Data Availability

All data generated or analyzed during this study are included in this published article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work of Drs. Lixin Wang and Jianhua Yang is supported by the National Security Agency NCAE-C grant H98230-20-1-0293 with Columbus State University, Georgia, USA.

References

[1] L. Wang, J. Yang, M. Mccormick, P.-J. Wan, and X. Xu, "Detect stepping-stone intrusion by mining network traffic using *k*-means clustering," in *39th IEEE International Performance Computing and Communications Conference (IEEE IPCCC 2020)*, November 2020.

[2] B. Mathew, "UNIX security: threats and solutions," in *Invited talk given at the 1995 system administration, networking, and security conference*, Washington, DC, April 1995.

[3] S. Staniford-Chen and L. T. Heberlein, "Holding intruders accountable on the Internet," in *Proceedings 1995 IEEE Symposium on Security and Privacy*, pp. 39–49, Oakland, CA, USA, 1995.

[4] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.

[5] L. Wang and J. Yang, "A research survey in stepping-stone intrusion detection," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, Article ID 1303, 15 pages, 2018.

[6] Y. Zhang and V. Paxson, "Detecting stepping-stones," in *Proceedings of the 9th USENIX Security Symposium*, pp. 67–81, Denver, CO, August 2000.

[7] J. Yang, B. Lee, and S. S.-H. Huang, "Monitoring network traffic to detect stepping-stone intrusion," in *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, pp. 56–61, Okinawa, Japan, March 2008.

[8] J. Yang, Y. Zhang, R. King, and T. Tolbert, "Sniffing and chaffing network traffic in stepping-stone intrusion detection," in *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 515–520, Krakow, May 2018.

[9] S. S. H. Huang, R. Lychev, and J. Yang, "Stepping-stone detection via request-response traffic analysis," in *4th IEEE International Conference on Automatic and Trusted Computing*, pp. 276–285, Hong Kong, China, July, 2007.

[10] S. S. H. Huang, H. Zhang, and M. Phay, "Detecting stepping-stone intruders by identifying crossover packets in SSH connections," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 1043–1050, Crans-Montana, 2016.

[11] T. He and L. Tong, "Detecting encrypted stepping-stone connections," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1612–1623, 2007.

[12] K. Yoda and H. Etoh, "Finding connection chain for tracing intruders," in *Proceedings of the 6th European Symposium on Research in Computer Security*, pp. 31–42, Toulouse, France, September 2000.

[13] A. Blum, D. Song, and S. Venkataraman, "Detection of interactive stepping-stones: algorithms and confidence bounds," in *Proceedings of International Symposium on Recent Advance in Intrusion Detection (RAID)*, pp. 20–35, Sophia Antipolis, France, September 2004.

[14] D. Bhattacherjee, "Stepping stone detection for tracing attack sources in software-defined networks," Degree Project in Electrical Engineering, Stockholm, Sweden, 2016.

[15] D. Donoho, A. Flesia, U. Shankar, V. Paxson, J. Coit, and S. Staniford, "Multiscale stepping-stone detection: detecting pairs of jittered interactive streams by exploiting maximum tolerable delay," in *5th International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science*, Berlin, Heidelberg, 2002.

[16] W. Ding, M. J. Hausknecht, S. H. S. Huang, and Z. Riggle, "Detecting stepping-stone intruders with long connection chains," in *2009 Fifth International Conference on Information Assurance and Security*, pp. 665–669, Xi'an, 2009.

[17] Xinyuan Wang and D. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by flow watermarking," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 434–449, 2011.

[18] Y. Chen and S. Wang, "A novel network flow watermark embedding model for efficient detection of stepping-stone intrusion based on entropy," in *Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE)*, 2016.

[19] K. H. Yung, "Detecting long connecting chains of interactive terminal sessions," in *Proceedings of International Symposium on Recent Advance in Intrusion Detection (RAID)*, pp. 1–16, Zurich, Switzerland, October 2002.

[20] J. Yang and S.-H. S. Huang, "A real-time algorithm to detect long connection chains of interactive terminal sessions," in *Proceedings of 3rd ACM International Conference on Information Security (Infosecu'04)*, pp. 198–203, Shanghai, China, November 2004.

[21] J. Yang and S. H. S. Huang, "Matching TCP packets and its application to the detection of long connection chains," in *Proceedings of 19th IEEE International Conference on Advanced Information Networking and Applications (AINA 2005)*, pp. 1005–1010, Taipei, Taiwan, China, March 2005.

[22] J. Yang and S. S.-H. Huang, "Mining TCP/IP packets to detect stepping-stone intrusion," *Computers & Security*, vol. 26, no. 7-8, pp. 479–484, 2007.

[23] G. Hamerly and J. Drake, "Accelerating Lloyd's algorithm for *k*-means clustering," in *Partitional Clustering Algorithms*, pp. 41–78, Springer, Cham, 2015.

[24] "Data clustering algorithms: *k*-means clustering algorithm," https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm.

[25] P. K. Agarwal and C. M. Procopiuc, "Exact and approximation algorithms for clustering," *Algorithmica*, vol. 33, no. 2, pp. 201–226, 2002.

[26] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient *k*-means clustering algorithm: analysis and implementation," *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881–892, 2002.

[27] J. Yang, L. Wang, A. Lesh, and B. Lockerbie, "Manipulating network traffic to evade stepping-stone intrusion detection," *Internet of Things*, vol. 3-4, pp. 34–45, 2018.

[28] J. Liu, W. Zhang, Z. Tang et al., "Adaptive intrusion detection via GA-GOGMM-based pattern learning with fuzzy rough set-based attribute selection," *Expert Systems with Applications*, vol. 139, article 112845, 2020.

[29] H. Clausen, M. S. Gibson, and D. Aspinall, "Evading stepping-stone detection with enough chaff," in *14th International Conference*, pp. 431–446, Melbourne, VIC, Australia, 2020.

[30] Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, "Towards query-efficient adversarial attacks against automatic speech recognition systems," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 896–908, 2021.

[31] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[32] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[33] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[34] Q. Li and D. L. Mills, "On the long-range dependence of packet round-trip delays in Internet," in *ICC'98. 1998 IEEE International Conference on Communications. Conference Record. Affiliated with SUPERCOMM'98 (Cat. No. 98CH36220)*, vol. 2, pp. 1185–1191, 1998.

[35] J. Yang, S.-H. S. Huang, and M. D. Wan, "A clustering-partitioning algorithm to find TCP packet round-trip time for intrusion detection," in *20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06)*, vol. 1, Vienna, 2006.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8891204, 8 pages
https://doi.org/10.1155/2021/8891204

Research Article

Detection Mechanisms of One-Pixel Attack

Peng Wang, Zhipeng Cai, Donghyun Kim, and Wei Li

Department of Computer Science, Georgia State University, Atlanta 30303, USA

Correspondence should be addressed to Zhipeng Cai; zcai@gsu.edu

Received 21 September 2020; Revised 22 December 2020; Accepted 3 February 2021; Published 23 February 2021

Academic Editor: Wenzhong Li

In recent years, a series of researches have revealed that the Deep Neural Network (DNN) is vulnerable to adversarial attack, and a number of attack methods have been proposed. Among those methods, an extremely sly type of attack named the one-pixel attack can mislead DNNs to misclassify an image via only modifying one pixel of the image, leading to severe security threats to DNN-based information systems. Currently, no method can really detect the one-pixel attack, for which the blank will be filled by this paper. This paper proposes two detection methods, including trigger detection and candidate detection. The trigger detection method analyzes the vulnerability of DNN models and gives the most suspected pixel that is modified by the one-pixel attack. The candidate detection method identifies a set of most suspected pixels using a differential evolution-based heuristic algorithm. The real-data experiments show that the trigger detection method has a detection success rate of 9.1%, and the candidate detection method achieves a detection success rate of 30.1%, which can validate the effectiveness of our methods.

1. Introduction

Deep learning is an artificial intelligence technique that follows the structure of a human's brain and imitates the neural cells in the human brain [1]. Over the past decades, deep learning has made significant progresses in speech recognition, natural language processing [2], computer vision [3], image classification [4], and privacy protection [5–7]. In particular, with the increase of data volume, traditional machine learning algorithms, such as SVM [8] and NB [9], suffer a performance bottleneck, in which adding more training data cannot really enhance their classification accuracy. Differently, the deep learning classifiers can continue to get improvements if more data is fed. However, it has been revealed that artificial perturbation can make the deep learning models misclassify easily. A number of effective methods have been proposed to produce so-called "adversarial samples" to fool the models [10, 11] and some work focused on fighting against adversarial attack [12–14].

1.1. One-Pixel Attack. Among the existing works, the one-pixel attack takes an extreme scenario into consideration, where only one pixel of an image is allowed to be modified to mislead the classification models of the Deep Neural Network (DNN) such that the perturbed image is classified to another label different from the image's original label [15]. As shown in Figure 1, with the modification of one pixel, the classification of images is changed to totally irrelevant labels.

The one-pixel attack is harmful to the performance guarantee of DNN-based information systems. Via modifying only one pixel in an image, the classification of the image may change to an irrelevant label, leading to performance degradation of DNN-based applications/services and even other serious consequences. For examples, in medical image systems, the one-pixel attack may make a doctor misjudge the disease of patients, and in autodriving vehicles, the one-pixel attack may cause serious traffic accidents on roads.

More importantly, the one-pixel attack is more threatening than other types of adversarial attack as it can be implemented easily and effectively to damage system security. Since the one-pixel attack is a type of black box attack, it does not require any additional information of the DNN models. In practice, the one-pixel attack only needs the probabilities of different labels instead of the inner information about the target DNN models, such as gradients and hyperparameters. The effectiveness of the one-pixel attack towards DNNs has been validated in [15], where the attack success rate is 31.40% on the original CIFAR-10 image dataset and

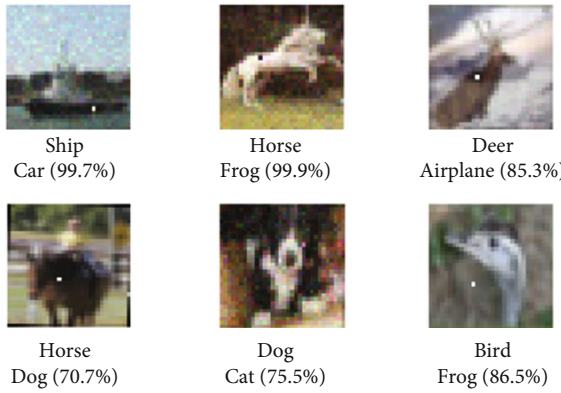


Figure 1: An example of one-pixel attack from [15].

16.04% on the ImageNet dataset. Such a success rate is large enough to break down an image classification system.

Therefore, to avoid the loss of system performance, detecting the one-pixel attack becomes an essential task.

1.2. Technical Challenges. The following two facts result in the difficulty of examining a one-pixel attack in images.

(1) *Extremely Small Modification*. The one-pixel attack modifies only one pixel in an image, which is significantly less than other types of adversarial attack. This makes the detection of the one-pixel attack very challenging.

(2) *Randomness of Pixel Modification*. For an image, there may be more than one feasible pixel that can cause the change of classification. In [15], the one-pixel attack randomly selects one of those feasible pixels to mislead the classifiers. Such randomness makes the detection of the one-pixel attack harder.

1.3. Contributions. In this paper, we develop two methods to detect a one-pixel attack for images, including trigger detection and candidate detection. In the trigger detection method, based on a concept named the "trigger" [16], we use gradient information of the distance between the pixels and target labels to find the pixel that is modified by the one-pixel attack. By considering the property of the one-pixel attack, in the candidate detection method, we design a differential evolution-based heuristic algorithm to find a set of candidate victim pixels that may contain the modified pixel. Intensive real-data experiments are well conducted to evaluate the performance of our two detection methods. To sum up, this paper has the following multifold contributions.

(i) To the best of our knowledge, this is the first work to study the detection of the one-pixel attack in literature, which can contribute to the defense of the one-pixel attack in future research

(ii) Two novel detection mechanisms are proposed, in which the modified pixels can be identified in two different ways based on our thorough analysis on the one-pixel attack

(iii) The results of real-data experiments validate that our two detection methods can effectively detect the one-pixel attack with satisfied detection success rates

The rest of this paper is organized as follows. In "Related Works," the existing works on adversarial attacks and detection schemes are briefly summarized. The attack model and the detection model are presented in "System Models." Our two detection methods are demonstrated in "Design of Detection Methods." After analyzing the performance of our methods in "Performance Validation," we conclude this paper and discuss our future work in "Conclusion and Future Work."

2. Related Works

A one-pixel attack is a special type of adversarial attack and is designed based on a differential evolution scheme. Thus, this section summarizes the most related literatures from the following two aspects: adversarial attack and detection of an adversarial attack.

2.1. Adversarial Attack. In an adversarial attack, attackers intend to mislead classifiers by constructing adversarial samples. Nguyen et al. made efforts on fooling a machine learning algorithm [17] and found that DNNs give high confidence results to random noise, which indicates that universal adversarial perturbation in a single crafted perturbation can cause a misclassification on multiple samples. In [10, 18], back-propagation is used to find gradient information of machine learning models, and gradient descent methods are used to build adversarial samples.

Since it might be hard or impossible to learn the internal information of a DNN model in practice, some approaches have been proposed to generate adversarial samples without using the internal characteristics of DNN models. Such approaches are called a black box attack [19–21]. Particularly, a special type of black box attack is the one-pixel attack, in which only one pixel is allowed to be modified. Under the one-pixel attack of [15], an algorithm was developed to find a feasible pixel for malicious modification based on differential evolution that has a higher probability of finding an expected solution compared with gradient-based optimization methods. Due to the concealed modification of only one pixel, it becomes more difficult to detect the one pixel attack. As mentioned in [15], the one-pixel attack requires only black box feedback that is the probability label without any inner information of the target network, like gradients or structure.

2.2. Detection of Adversarial Attack. On the other hand, research attention is also paid to work out detection methods for adversarial attack. Papernot et al. provided a comprehensive investigation into the security problems of machine learning and deep learning, in which they established a threat model and presented the "no free lunch" theory showing the tradeoff between accuracy and robustness of deep learning models. Inspired from the fact that most of the current datasets are compressed JPG images, some researchers designed a method to defend image adversarial attack using image compression. However, in their proposed method, a large

compression may also lead to a large loss of classification accuracy of the attacked images, while a small compression cannot work well against adversarial attack. In [16], Neural Cleanse was developed to detect a backdoor attack in neural networks, and some methods were designed to mitigate backdoor attack as well.

Compared with the existing works, this paper is the first work that focuses on the detection of the one-pixel attack. In particular, two novel detection mechanisms are proposed with one using a gradient calculation-based method and the other using a differential evolution-based method.

3. System Models

The attack model and detection model in our considered DNN-based information systems are introduced as follows.

3.1. Attack Model. In this paper, the attack model of [15] is taken into account, in which an adversarial image is generated by modifying only one pixel in the victim image. The purpose of a one-pixel attack is to maliciously change the classification result of a victim image from its original label to a target label. As shown in Figure 2, the image is correctly classified as an original label, "sheep," by a given DNN model. After being modified one pixel, the output label with the highest preference of the model is changed to a target label, "airplane," leading to a wrong classification result. The attackers perform a black box attack only, which means they have the accessibility to the probability labels and cannot get the inner information of the network. Also, considering that the attacker aims to make the attack as efficiently as possible, it is supposed that all the images in the dataset are altered.

3.2. Detection Model. Suppose that a set of adversarial images, which are modified by the aforementioned one-pixel attack, are given to the system. With these given images, our objective is to distinguish which pixel has been modified by a one-pixel attack.

To detect the one pixel attack, two novel methods are developed. The first method is the "trigger detection" to identify the modified pixel, and the second one is the "candidate detection" to find a set of victim pixels. The "trigger detection" model is designed for white box detection that requires all the network information including inner gradients and network structure. In the trigger detection model, we first propose a new concept named "trigger" for image data and then detect the trigger in a given adversarial image. If the detected trigger is the pixel modified by the one-pixel attack, our detection is successful. The "candidate detection" is for the black box detection, where only the output probabilities of labels are needed for the detection. In the candidate detection model, we aim to find a set of pixels as the candidate victim pixels. If the selected victim pixels include the pixel modified by the one-pixel attack, our detection is successful. The details of our two detection mechanisms are demonstrated in "Design of Detection Methods."

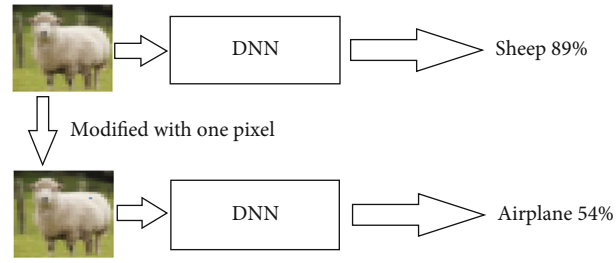


Figure 2: Illustration of one-pixel attack.

4. Design of Detection Methods

Our proposed detection methods are described as follows. For a better presentation, the major notations are summarized in Table 1.

4.1. Trigger Detection Method

4.1.1. Main Steps. Formally, the trigger of an image is defined to be the pixel that has the greatest impact on the model classification. Thus, any image, which has a properly modified trigger, should have a higher confidence on a target label and will be likely to be classified as the target label regardless of other unchanged image features. In other words, the classification result would be changed to a target label if the trigger is modified properly using DNNs' properties.

Let $\mathbb{L}$ represent the set of output labels in a DNN model. For any adversarial image, we assume its original label is $L_{\text{in}} \in \mathbb{L}$ and its target label is $L_t \in \mathbb{L}$, where in $\neq t$. Under a one-pixel attack model, the modification on the trigger pixel can transform all inputs of L_{in} to be classified as L_t.

Motivated by the above observations, the one-pixel attack can be detected via identifying the triggers of adversarial images according to the following steps.

Step 1. For a given label, we treat it as a potential target label in the one-pixel attack. We use an optimization-based scheme to find the trigger that can misclassify all samples from other labels into the target label.

Step 2. We repeat Step 1 for each output label in the DNN model and obtain N potential triggers, in which $N = |\mathbb{L}|$ is the number of labels of the DNN model.

Step 3. After calculating N potential triggers, we measure the size of each trigger. The size of a trigger is defined to be the modified RGB value of the trigger pixel. Then, the outlier detection algorithm of [22] is adopted to detect whether the perturbation of each potential trigger is significantly smaller than others. A significant outlier is likely to indicate a real trigger, and the label matching the real trigger is the target label in the one-pixel attack.

4.1.2. Trigger Identification. The details of our optimization-based scheme for identifying trigger are addressed below.

Table 1: Notations.

Notations	Meaning
$\mathbb{L}$	Output labels in a DNN model
L_{in}	The original label
L_t	The target label of a one-pixel attack
$A(\bullet)$	The function that applies a trigger to an image
x	The original image
x'	The modified image A pixel of x and $x_{i,j,c}$ represents a pixel of x'
$x_{i,j,c}$	i and j are the x and y coordinates of the pixel, respectively, and c is the color channel
Δ	A trigger of an image
$\text{Loss}(\bullet)$	The loss function measuring classification error
x_{next}	An element of the candidate solution
$T_{\max}$	Maximum iteration numbers

Suppose X is a set of clean images without modification. A generic form of injecting trigger for any original image, $x \in X$, is given in

$$A(x, m, \Delta) = x', \tag{1}$$

where $A(\cdot)$ represents the function that applies a trigger to x. Correspondingly, the modified image is denoted as x'. Let $x_{i,j,c}$ represent a pixel of x and $x'_{i,j,c}$ represent a pixel of x', where i and j are the x and y coordinates of the pixel, respectively, and c is the color channel. The relationship between x and x' can be mathematically expressed by

$$x'_{i,j,c} = (1 - m) \cdot x_{i,j,c} + m \cdot \Delta_{i,j,c}, \tag{2}$$

in which Δ represents a trigger of x and $m \in [0, 1]$ describes how much Δ can overwrite the original image. Particularly, when $m = 1$, the pixel of the trigger completely overwrites the original color, and when $m = 0$, the original color is not modified at all. The original image x is classified to the original label L_{in}, and the modified image x' is classified to the target label L_t.

Then, given the target label L_t, the problem of finding a trigger can be formulated as a multiobjective optimization problem, i.e.,

$$\min_{m,\Delta} \text{Loss}(L_t, f(A(x, m, \Delta))) + m, \quad \forall x \in X. \tag{3}$$

In Eq. (3), $\text{Loss}(\cdot)$ is the loss function measuring classification error that is computed by cross entropy and $f(\cdot)$ is the prediction function of the DNN model.

In this paper, L1 norm of m is adopted to measure the magnitude of the trigger. By solving the above optimization problem, we get the trigger, Δ, for each target label and its L1 norm. Next, in Step 3, we identify the triggers that show up as outliers with smaller L1 norm by utilizing the outlier detection algorithm of [22].

4.2. Candidate Detection Method. Notably, to generate adversarial images, the one-pixel attack of [15] uses a differential evolution algorithm to randomly select a pixel that can lead misclassification on an image. Mathematically speaking, for the problem of image generation, the selected pixel is a feasible solution and may not be an optimal solution. Therefore, the obtained trigger pixel might not be actually modified in the one-pixel attack, resulting in failed detection. In order to improve the attack detection success rate, the goal of our candidate detection method is to find a set of victim pixels (i.e., a set of feasible solutions), each of which satisfies the requirement of adversarial image generation in the one-pixel attack.

4.2.1. Problem Formulation. Without loss of generality, we assume an input image can be represented by a vector in which each scalar element represents one pixel. In a DNN model, $f(\cdot)$ receives an image as input and gives confidence of N labels. Accordingly, the probability of x being classified to a label $L \in \mathbb{L}$ is $f(x)[L]$. For an original image x, an additive adversarial perturbation is represented by a vector v. The modification degree is measured by the length of v, and the allowable maximum modification is 1 in the one-pixel attack model. The problem of generating adversarial images using the one-pixel attack is formulated as follows.

$$\begin{aligned} \max_{\Delta} \quad & f(x + \Delta)[L], \\ \text{s.t.} \quad & \|\Delta\| \le 1. \end{aligned} \tag{4}$$

4.2.2. Differential Evolution-Based Heuristic Algorithm. To obtain the solution to Equation (4), a heuristic algorithm is designed based on differential evolution (DE), which brings the following benefits.

(i) DE gives a higher probability of finding global optimal solutions as well as a lower probability of getting "trapped" in the local solutions compared with gradient descent and greedy searching algorithms

Input: an adversarial image generated by a one-pixel attack, a DNN classifier, and a label set $\mathbb{L}$ with $|\mathbb{L}| = N$
Output: a set of pixels containing C candidate victim pixels
1: Initialize model with the target DNN model
2: Randomly chose N_{ini} pixels from the image, and for each point, randomly set a color as present pixel set
3: Calculate the confidence of the image on all N labels
4: **for all** each label $L \in \mathbb{L}$ **do**
5: **while** true **do**
6: Calculate the change of the confidence on L when modifying with parent pixels
7: Generate offspring pixels based on Equation (5)
8: Calculate the change of confidence on L when modifying with offspring pixels
9: Select N_c pixels with the highest confidence changed as new parent pixels
10: **if** all top C confidences changed on targets are larger than p_{th} **then**
11: Save the top C pixels as candidate victim pixels
12: Set AttackSucc = true
13: Break while loop
14: **end if**
15: **if** while loop is over $T_{\max}$ times **then**
16: Break while loop
17: **end if**
18: **end while**
19: **end for**
20: **if** AttackSucc is true **then**
21: **return** C candidate victim pixels
22: **else**
23: **return** fail to find candidates
24: **end if**

Algorithm 1 Algorithm of candidate detection method.

(ii) DE requires less information from the optimization objectives. DE does not require gradient information from the dataset, which means it even does not require the problem to be differentiable. Under the extremely strict constraint of modifying only one pixel in an image, the problem is not differentiable and can be effectively resolved by DE

(iii) To detect a one-pixel attack, we only need to know whether the confidence changes after modifying a pixel, which can be formulated and solved in a simple way using DE

In this paper, we encode the perturbation into an array (i.e., a candidate solution) which is optimized (evolved) by differential evolution. One candidate solution contains a fixed number of perturbations, and each perturbation is a tuple holding five elements including x-y coordinates and RGB value of the perturbation, where one perturbation modifies one pixel. The DE algorithm is performed iteratively and is terminated when one of the two conditions is satisfied: (i) the maximum number of iteration $T_{\max}$ is reached or (ii) the probability of being classified to the target label exceeds a threshold p_{th}. Let N_{ini} be the initial number of candidate solutions (population) and N_c be the number of candidate solutions (i.e., children) produced in each iteration. At the $(g+1)$-th iteration, N_c candidate solutions are produced from the g-th iteration via the following DE formula:

$$x_{\text{next}}(g+1) = x_{r_1}(g) + F\left(x_{r_2}(g) - x_{r_3}(g)\right), \tag{5}$$

where x_{next} is an element of the candidate solution; r_1, r_2, and r_3 are random values with $r_1 \neq r_2 \neq r_3$; and F is the scale parameter. After being generated, each candidate solution competes with their parents according to the index of the population, and the winners survive in the next iteration. When the algorithm terminates, C candidates are output.

The pseudocode of our algorithm is shown in Algorithm 1. For each image, the above algorithm will go through all the N labels, i.e., the "for loop" in lines 4-19 of Algorithm 1. For each label, the candidate selection process will run up to $T_{\max}$ iterations, i.e., the "while loop" in lines 5-18 of Algorithm 1. Each iteration costs a constant time to generate N_c children and pick N_c winners. As a result, the time complexity of Algorithm 1 is $O(N \cdot T_{\max})$.

5. Performance Validation

In this section, extensive real-data experiments are conducted to evaluate the performance of our two detection methods.

5.1. Experiment Settings. Our experiments adopt CIFAR-10 as the dataset and VGG-16 as the DNN model. Table 2 shows the structure of the VGG-16 network which is the same as the network used in a one-pixel attack. After training, we get the model with the accuracy as shown in Table 3.

To measure the performance of the one-pixel attack, we calculate the classification accuracy of the VGG-16 model on the test images. Also, we calculate the success rate of launching the one-pixel attack, in which "airplane" is set as

Table 2: Network structure of VGG-16.

Conv2d layer (kernel = 3, stride = 1, depth = 64)
Conv2d layer (kernel = 3, stride = 1, depth = 64)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 128)
Conv2d layer (kernel = 3, stride = 1, depth = 128)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 256)
Conv2d layer (kernel = 3, stride = 1, depth = 256)
Conv2d layer (kernel = 3, stride = 1, depth = 256)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Max pooling layer (kernel = 2, stride = 2)
Flatten layer
Fully connected (size = 2048)
Fully connected (size = 2048)
Softmax classifier

Table 3: Classification accuracy of VGG-16.

Model	Accuracy on test set	Claimed accuracy on test set
VGG-16	93.4%	94%

Table 4: Success rate of one-pixel attack.

Model	Accuracy on test set	Claimed accuracy on test set
VGG-16	93.4% ± 1.65%	17.3% ± 3.61%

the target label. The success of the one-pixel attack means after being modified a pixel, an image whose original classified label is not airplane is misclassified to airplane by the DNN network. Moreover, to reduce the influence caused by the randomness in the differential evolution algorithm of the one-pixel attack, the classification process is repeated 5 times, and the average accuracies and their variances are presented in Table 4.

Specifically, the one-pixel attack is launched towards 60,000 testing images and succeeds in making 8655 nonairplane images get classified to airplane. In the experiments, we pick 8000 such adversarial images to evaluate our detection method.

5.2. Performance Metrics. The performance metrics are introduced as follows.

(i) *Label confidence*. The label confidence is the confidence of different labels given by the DNN model to an image. For a label, higher confidence means a higher probability that the image is classified to the label.

(ii) *Detection success rate*. The definition of successful attack detection is different in our two detection methods. In trigger detection, our detection is successful if the detected trigger is the pixel modified by a one-pixel attack, while, in the candidate detection model, our detection is successful if the pixel modified by the one-pixel attack is included in the set of selected victim pixels. For both detection methods, the detection success rate is defined as the ratio of the number of successful detection to the number of adversarial images.

5.3. Performance of Trigger Detection. Since the L1 norm has a better feature selection performance and interpretability [23], our trigger detection method uses the L1 norm to measure the distance between the pixel and a label. If the L1 norm of a pixel is obviously different from others, we can consider that the pixel is infected. Figure 3 shows the L1 norm of all pixels of an infected image.

From Figure 3, one can find that the L1 norm of the 751st pixel is obviously different from others. With verification, we know that the 751st pixel is the pixel modified in the one-pixel attack. Also, to understand how the affected target label is related to the modified pixel, we calculate the L1 norm of the infected pixel to different labels. In Figure 4, we can find that the L1 norm to the airplane is lower than that to the other labels. Thus, our approach can also determine which label is the target label.

The average detection success rate of our trigger detection method is 9.1%.

5.4. Performance of Candidate Detection. In our candidate detection method, the initial number of candidate solutions and the number of produced candidate solutions are set to be 400, i.e., $N_{\text{ini}} = N_c = 400$; the maximum number of iterations is $T_{\max} = 100$; the scale parameter is $F = 0.5$; and the threshold for the probability of being classified to the target label is $p_{\text{th}} = 90\%$. To eliminate the influence of random variables in our Algorithm 1, we run the experiment 5 times with the fixed parameter settings and present the results in Table 5. Moreover, to investigate the impact of the size of candidate set, we also compare the detection success rates when C is set to be different values.

As shown in Figure 5, when $C = 1$, the success detection rate is 5.4% smaller than the success detection rate of our trigger detection method. Particularly, with $C = 1$, both the trigger and the candidate detection methods output one detected pixel but differ in their pixel selection schemes: (i) in out trigger detection, the trigger pixel is an optimal solution of trigger identification problem as well as has a smallest value of the L1 form, while (ii) in our candidate detection, the only one candidate victim pixel is randomly selected by the differential evolution-based heuristic algorithm. From the definition of the trigger pixel in this paper, the probability of the trigger pixel being modified in a one-pixel attack is larger than that of other pixels. Therefore, the trigger

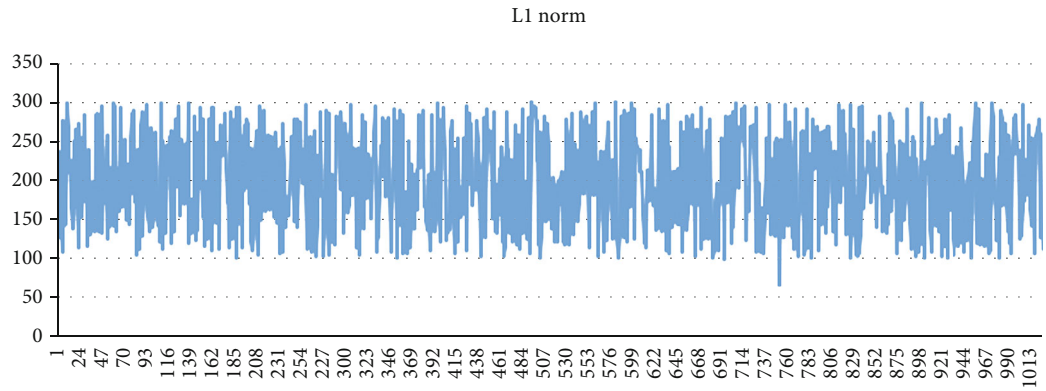


Figure 3: L1 norm of an infected image to label airplane.

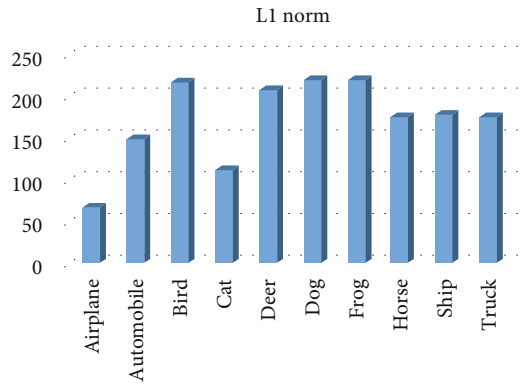


Figure 4: L1 norm of an infected image to airplane label.

Table 5: Detection success rate of candidate detection.

Experiment	1	2	3	4	5
Success rate (%)	20.4	24.3	30.1	21.9	26.7

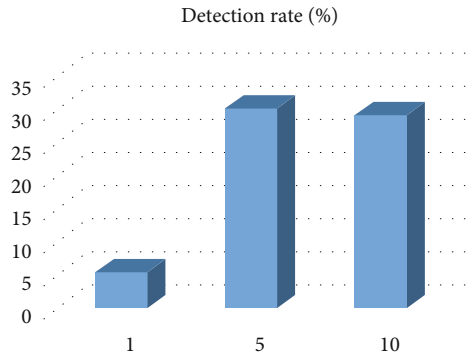


Figure 5: Impacts of the number of output candidates.

detection method outperforms the candidate detection method when $C = 1$, confirming that the idea of finding the trigger pixel to detect the one-pixel attack is solid.

When C is increased from 1 to 5, the detection success rate grows from 5.4% to 30.1%. The main reason is that with a larger number of output candidate pixels, more possible modified pixels can be examined, and the probability of finding the actual modified pixel is increased.

However, when C is increased from 5 to 10, the detection success rate nearly remains the same (in Figure 5, the subtle difference of the detection success rate results from the randomness of the DE algorithm). This illustrates that only increasing the number of candidate pixels may not always enhance the detection success rate. In summary, for the detection success rate, the marginal benefit of enlarging the number of candidate pixels is diminishing, and thus, setting an appropriate value to C can help effectively and efficiently detect the one-pixel attack (e.g., $C = 5$ in our experiments).

6. Conclusion and Future Work

This paper proposes two novel methods, i.e., the trigger detection method and the candidate detection method, to detect a one-pixel attack that is one of the most concealed attack models. The trigger detection method gives the exact pixel that may be modified by the one-pixel attack; the candidate detection method outputs a set of pixels that may be changed in the one-pixel attack. Via extensive real-data experiments, the effectiveness of our two methods can be confirmed; in particular, the detection success rate of our candidate detection can achieve 30.1%.

As a preliminary exploration of the one-pixel attack detection, in this paper, we consider that all the images are attacked and the detection is thus implemented on a dataset full of modified images. In our future work, we will carry out further research activities along two directions: (i) attempting to distinguish between the benign images and the attacked images in the presence of the one-pixel attack and (ii) mitigating the impact of the one-pixel attack by enhancing the resistance to adversarial samples in DNNs.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by the National Science Foundation of the U.S. (1704287, 1829674, 1912753, and 2011845).

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 129–136, Bellevue, Washington, USA, 2011.

[3] A. Kendall and Y. Gal, *What uncertainties do we need in Bayesian deep learning for computer vision?*, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 2017.

[4] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: a simple deep learning baseline for image classification?," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[5] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[6] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[7] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–590, 2016.

[8] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer science & business media, 2013.

[9] D. D. Lewis, "Naive (Bayes) at forty: the independence assumption in information retrieval," in *European conference on machine learning*, pp. 4–15, Springer, 1998.

[10] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, Saarbrucken, Germany, March 2016.

[11] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[12] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[13] Z. Xiong, Z. Cai, Q. Han, A. Alrawais, and W. Li, "Adgan: protect your location privacy in camera data of auto-driving vehicles," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[14] Z. Xiong, W. Li, Q. Han, and Z. Cai, "Privacy-preserving auto-driving: a GAN-based approach to protect vehicular camera data," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 668–677, Beijing, China, November 2019.

[15] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[16] B. Wang, Y. Yao, S. Shan et al., "Neural cleanse: identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723, San Francisco, CA, USA, May 2019.

[17] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, Boston, MA, USA, 2015.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, Las Vegas, NV, USA, 2016.

[19] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, Honolulu, HI, USA, 2017.

[20] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, Abu Dhabi, United Arab Emirates, April 2017.

[21] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56, San Francisco, CA, USA, May 2018.

[22] I. Ben-Gal, "Outlier detection," in *Data mining and knowledge discovery handbook*, pp. 131–146, Springer, 2005.

[23] S. Wu, G. Li, L. Deng et al., "$L1$-norm batch normalization for efficient training of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2043–2051, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6643229, 15 pages
https://doi.org/10.1155/2021/6643229

Research Article

A Semiopportunistic Task Allocation Framework for Mobile Crowdsensing with Deep Learning

Zhenzhen Xie,[1] **Liang Hu**,[1] **Yan Huang**,[2] **and Junjie Pang**[3]

[1]*College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China*
[2]*College of Computing and Software Engineering at Kennesaw State University, Atlanta, GA, USA*
[3]*Business School and the College of Computer Science and Software Engineering, Qingdao University, Qingdao, Shandong 266000, China*

Correspondence should be addressed to Yan Huang; yhuang24@kennesaw.edu

Received 13 October 2020; Revised 22 December 2020; Accepted 1 February 2021; Published 16 February 2021

Academic Editor: Xiao Zhang

The IoT era observes the increasing demand for data to support various applications and services. The Mobile Crowdsensing (MCS) system then emerged. By utilizing the hybrid intelligence of humans and sensors, it is significantly beneficial to keep collecting high-quality sensing data for all kinds of IoT applications, such as environmental monitoring, intelligent healthcare services, and traffic management. However, the service quality of MCS systems relies on a dedicated designed task allocation framework, which needs to consider the participant resource bottleneck and system utility at the same time. Recent studies tend to use a different solution to solve the two challenges. The incentive mechanism is for resolving the participant shortage problem, and task assignment methods are studied to find the best match of participants and system utility goal of MCS. Thus, existing task allocation frameworks fail to consider the participant's expectations deeply. We propose a semiopportunistic concept-based solution to overcome this issue. Similar to the "shared mobility" concept, our proposed task allocation framework can offer the participants routing advice without disturbing their original travel plan. The participant can accomplish the sensing request on his route. We further consider the system constraints to determine a subgroup of participants that can obtain the utility optimization goal. Specifically, we use the Graph Attention Network (GAT) to produce the target sensing area's virtual representation and provide the participant with a payoff-maximized route. Such a method makes our solution adapt to most of MCS scenarios' conditions instead of using fixed system settings. Then, a reinforcement learning- (RL-) based task assignment is adopted, which can help the MCS system towards better performance improvements while support different utility functions. The simulation results on various conditions demonstrate the superior performance of the proposed solution.

1. Introduction

The Mobile Crowdsensing (MCS) paradigm, as a crucial part of the current IoT ecosystem, is offering an integrated sensing capability for various aspects of the human environment, such as urban environmental monitoring [1], traffic management [2], and intelligent healthcare applications [3]. Taking advantage of the essential features of "crowdsourcing-based sensing," existing IoT applications and platforms have significantly relieved sensor resources' poverty by extending the sensing range and various sensing capabilities to support more types of sensing tasks [4]. Compared to other sensing data collection methods, we observe a significant difference between traditional sensing service and MCS systems: rather than deploying specified sensors for IoT applications with expensive cost, MCS leverages the sensing capability of multiple types of mobile devices to build a "human-in-the-loop" system [5]. In this system, each participant could be a potential worker to accomplish the sensing tasks. When they accept the task request, the participants would follow the task guidance and use the personal devices (like phones, vehicles, or smart home devices) as the sensors to collect data and then upload them to the MCS platform for further processing.

To implement this working process, MCS platforms need to select participants and provide effective rewards to make participants keep high-quality contributions. The task

allocation problem is thus essential for encouraging people to participate proactively in MCS tasks, while scheduling the sensing resource and requests properly [6]. There are two primary challenges that need to be solved for the MCS task assignment problem: the first challenge is how to accumulate and maintain a sufficient large participant pool, which refers to the participant resource bottleneck [7]; the second one is how to maximize the MCS platform benefits or satisfy task requirements under limited system budget constraints, which refers to the budget-utility contradiction [8].

In coping with the above challenges, most of the existing works assume the MCS systems already have an accurate trajectory prediction method based on sufficient mobility data and valid incentive settings. Then, the "opportunistic mode" or "participatory mode" is adopted as the primary method to organize the participants and distribute the tasks. In the opportunistic mode, the MCS platforms manage the sensing request, predict each potential worker's future trajectory, and finally distribute the sensing request to the workers who have similar routes. Thus, the workers can keep their daily routine since the sensing tasks would not disturb their original plan. Alternatively, the participatory mode means that the MCS platform needs to schedule the participants and task requests; meanwhile, the participants should follow the route to accomplish the sensing request. When they accept the sensing task, they might change their original daily routine or trip plan to follow the planned route and collect sensing data.

However, both of the two modes have several limitations. The opportunistic mode implies an essentially passive framework; the performance of the MCS platform is bounded with the conditions of participants' historical mobility and trajectory prediction method [9]. Furthermore, the MCS platform needs enough time to accumulate participant trajectory information to keep the prediction method working steadily. Furthermore, the participants' expected payoff in this mode could be high since they need to expose more historical trajectories to help the MCS platform understand their daily routines for better task scheduling, which may significantly raise privacy concerns [10]. Differently, the participatory mode uses an active way to schedule the participants and requests, but it may fail to satisfy the individual expectation of participants, especially when they are not prepared to change their original travel plan. Besides, it is notable that although the opportunistic mode considers not disturbing the participants' daily routine, they fail to provide a route to maximize the participants' payoff. As we know, unsatisfactory income can broadly impact the motivations to keep uploading high-quality data [11].

We also observe that the above limitations could be exacerbated when MCS platforms exhibit increasing sensing requests that uncertainly occur. For example, urban emergencies in healthcare and traffic scheduling [12] often need information for some targeted areas with no forecasts. These timely sensing requests come in randomly with a very short life cycle, and there is no explicit recurring need for collecting data. The sensing resource can hardly be prepared well in advance so that they enlarge the above contradiction between the increasing demand of sensing requests and the limited participant resources.

To fill the research gap, we expect to use an effortless but payoff-maximized route to motivate the participant to accomplish sensing requests. Thus, in this work, we propose a novel semiopportunistic task allocation design. Different from the existing solution, we allow participants to send the travel plan to the MCS platform, including the start and end addresses and also time constraints. Our proposed method would actively provide routing advice to maximize each participant's payoff. Compared to the task assignment adopting the opportunistic or participatory mode, our proposed semiopportunistic task allocation concept has a similar property like payoff guarantee but less impact on the participants' original travel plan. For clarity, we give an example scenario illustrated in Figure 1, and we assume that the MCS platform has seven sensing requests and five potential participants with a fixed starting point and ending point. Then, our proposed framework could provide routing advice with a maximized payoff and finally select three participants as the workers to achieve the MCS platform's performance goal.

Therefore, our work contains two parts:

(1) *Semiopportunistic sensing-based participant profiling* is aimed at resolving the participant resource bottleneck; we introduce the novel semiopportunistic sensing concept into our solution, which could be regarded as the combination of the opportunistic mode and participatory mode. This semiopportunistic sensing concept is inspired by the "shared mobility" idea [13]. The popularity of "shared mobility" gives us a novel perspective to rethink the MCS participant's bottleneck challenge. We investigated that pilot studies on real-world applications prove that "shared mobility" offers several benefits for our environmental concerns like reduced emissions and traffic congestion. Meanwhile, participants of such applications can share costs of travel or earn additional income by accepting several route changes [14, 15]

Motivated by these successful attempts, our design is aimed at adopting a similar perspective to accumulate potential participators and promote their willingness to accomplish the tasks. Specifically, unlike the opportunistic sensing method using workers to complete tasks unintentionally, we plan the participant's trip with more dedicated sensing request selection: our proposed method could insert the sensing request to their trips with no harm to their origin-destination stations and consider the individual time constraints but ensure the planned routes with a maximized accumulated payoff.

(2) *Reinforcement learning-based participant selection* is aimed at considering the system budget constraints and maximizing the utility of MCS platforms; we need to decide on a subgroup of participants to optimize the system utility under different settings. Furthermore, we expect the solution to automatically adapt to large-scale participants and choose the optimal mapping between tasks and participants under

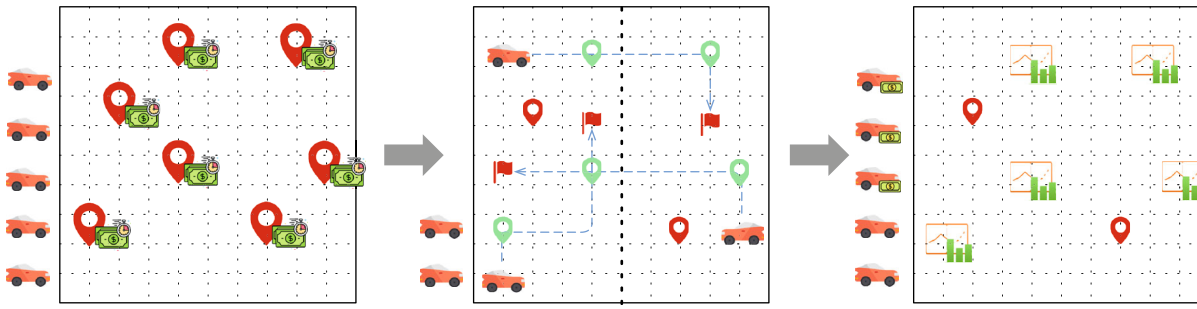


Figure 1: Framework overview.

different system constraints. Thus, we utilize the Double Deep Q-Network (DDQN) algorithm from the reinforcement learning theory to achieve an improved trade-off between stability and reactivity. It utilizes a goal-directed learning method to automatically gather the information about current participant profiles, system budget constraints, and system utility status and investigate a task allocation strategy by searching a group of participants that can guarantee the optimization goal

The fundamental differences from existing research provide the main contribution of this paper.

First, to adapt to individual participant willingness, we draw inspiration from the literature on the shared mobility concept to generate a quasi-optimal route for each worker to guarantee the maximum accumulated payoff on their daily trips.

Second, we employ Graph Attention Network techniques from graph representation learning so that the MCS platform can provide routing advice based on an improved understanding of the target sensing area. The separation of participant profiling and participant selection in our proposed framework is more tractable since the GAT method can simultaneously fuse the sensing request representations and the sensing area topology. Also, it can provide high-quality input of the participant selection method to simplify the searching process.

Finally, a participant selection strategy that utilizes the DDQN algorithm from the reinforcement learning theory is proposed to achieve an improved trade-off between stability and reactivity. It is worth mentioning the reward settings in DDQN-based participant selection decoupling the system utility goal of MCS platforms and the selection strategy, which suggests our participant selection method can adapt to different requirements for various MCS platforms.

The rest of the paper is organized as follows. Section 2 introduces related works. The system model of our proposed task allocation framework is presented in Section 3. Section 4 explains the implementation details. The simulations and results are analyzed in Section 5. Finally, the conclusion is presented in Section 6.

2. Related Works

Due to the rapid growth of big IoT data [16], the concept of MCS has been proposed to facilitate innovation in IoT sensing solutions. In various IoT scenarios like air pollution monitoring systems, urban traffic control, and noise monitoring, the MCS system offers these applications various types of sensing capability by a novel combination between humans and mobile devices. Furthermore, it can also provide plenty of information for various virtual services like information inferences [17], map services [18], and location-based social networks [19]. Task allocation becomes the major challenge to support such an MCS system since it directly affects the task response rate, sensing data quality, and economic benefit [20, 21]. This section reviews this area's recent progress and gives a brief introduction to several techniques related to our work.

2.1. Task Allocation Problem in Mobile Crowdsensing. The task allocation is an essential part of managing sensing requests and scheduling the sensing resource in MCS systems. A typical task allocation process has three primary steps: first, it characterizes current task requests under a limited system budget; second, the worker's mobility profiling with trajectory prediction or route is utilized to obtain the task acceptance rate; and finally, the platform selects a subset from existing workers to meet the system utility maximization goal like system budget, coverage ratio, or time cost of task execution.

Besides, to respond to the sensing request efficiently, existing task allocation schemes are bounded with several performance objectives, such as response latency, sensing coverage, and sensing quality of collected data. Thus, for implementing the task allocation scheme properly, recent research regards the following two problems as the primary challenges: first, select a group of participants with the maximum sensing coverage while still under the total system budget; second, translate the sensing tasks' needs into a system utility optimization goal precisely. For the first question, Zhang et al. [22] propose a coverage-oriented participant selection method and implement it as a searching process to reach the spatial coverage goal. For the latter, there are a lot of works that adopt similar thoughts with different approaches, like using enhanced greedy algorithms [7] or Genetic Algorithms (GA) [23]. Furthermore, there are also several works that consider the security concerns like sensitive information inference attacks and privacy leakage risks in the MCS task assignment [24]. For the second question, we observe some research studies are with different settings of the system utility goal. For example, Xiong et al. [25]

propose an iCrowd task allocation framework with a k-depth utility goal using a more individualized way to capture the coverage ratio of different tasks. They consider every task's spatial-temporal coverage needs and calculate the coverage by a flexible K threshold. Under such settings, the sensing resource will not be wasted on sensing tasks that cannot have enough participants.

However, we observe that most of the existing works are platform-oriented, which indicates that they consider more the platform's utility rather than the expectations of participants [26]. As a complementary, several incentive mechanisms are proposed to resolve the participant resource bottleneck by exploring multiple methods to motivate them to upload high-quality data with reasonable payoff settings [27–29], using the social network propagation theory [30] to support crowdsensing, so that they can investigate the preferences of participants to ensure the task acceptance rate. Besides, utilizing the enhanced privacy method to maintain the participant tool is another promising way, since more and more participants have privacy concerns when uploading data for IoT applications [31]. By using the influence propagation method, considering the privacy concerns of potential participants, or utilizing the recommender system-like concepts, such task assignment strategy can focus more on individual requirements to accumulate qualified workers actively [32]. There are also other types of works that consider designing a more secure MCS platform to motivate more workers to accomplish the tasks like using the blockchain-based framework [33]. Obviously, these works shift the focus from platform-oriented performance optimization to solving the trade-off between individual intentions and platform requirements.

It still faces several challenges in practical conditions; for example, the influence propagation-based task assignment needs to investigate participants' preference first, which is an additional cost that most of the applications are often unwilling to afford. What is more, these methods need much more private information than traditional approaches to meet the requirement to select the valid seed or construct the social graph.

To overcome these limitations, recent works use the Sparse Mobile Crowdsensing concept [34] instead to resolve the contradiction between the limited participant resource and the increasing need for sensing data volume. Such works use fewer participants as seed workers to collect raw data and then use data inference techniques to generate supplemental data. However, when several participants cannot provide sufficient sensing data, the sensing quality will be significantly affected.

Unlike the prior works, we propose a novel task allocation approach that uses semiopportunistic sensing to motivate potential participants to join the sensing task. The proposed method using the "shared mobility" idea reduces the extra expense for participants and provides routing advice that can further fulfill the sensing task requirements.

2.2. Graph Attention Networks. Representation learning for graph structure data is an emerging topic, and several novel neural networks are proposed to obtain graph embedding with low-dimensional representations of nodes. The Graph Attention Network (GAT) is recently regarded as a useful graph convolutional network architecture, which leverages the graph structure by the attention mechanism to extract intermediate feature representations for the nodes in the graph [35]. The sole layer of GAT is the graph attention layer, which accepts the input of a set of node features and outputs the node embedding results by performing self-attention computation between the target node and its first-order neighbors. Then, GAT uses the normalized attention coefficients to compute a linear combination of the neighbors' features and generate the final output embedding results [36]. Additionally, multihead attention is introduced in the self-attention learning process for stability consideration. By adopting a graph convolutional layer based on the masked attention mechanism, GAT satisfies several distinguished properties such as computational efficiency, considering each neighbor node's different importance, and better applicability to inductive learning problems. The interest in applying GAT to graph embedding has dramatically increased, including the traffic prediction [37], recommender system [38], and compliment techniques for knowledge graph completion [39]. Thus, we consider adopting GAT in the participant profiling step to generate routing advice for potential participants concerning different sensing area topology scales robust to the topology changes.

Reinforcement learning (RL) [40] has been used for a variety of learning tasks, ranging from resource allocation in IoT application scenarios [41], game AI like AlphaGo [42], and combinatorial optimization problems [43]. In typical settings of a model-based RL, the RL agent achieves the environment exploration and obtains an optimal policy to interact with the environment to maximize its benefits. Such a process is referred to as the Markov Decision Process (MDP), which explains an RL agent's life cycle by state, action, and reward. With the further support of deep learning, RL has proven its effectiveness on the problems requiring discrete stochastic control or continuous control with a significantly large sample size. It also shows much better performance than heuristic-based approaches. Multiple successful research attempts show that RL makes the system learn to manage the functions or resources by self-cognitive capability, which motivates us to design a feasible solution using RL to resolve the MCS task allocation problem. To the best of our knowledge, we are among the first to leverage the RL approach for enabling semiopportunistic sensing in MCS. With RL support, our solution is aimed at providing a practical self-management approach that can be easily extended to other task allocation problems with different system utility settings.

3. Problem Analysis and Formulation

This section gives the preliminary knowledge of typical task allocation problems in MCS, introduces our proposed semi-opportunistic concept as a novel solution for this problem, and explains the necessity and advantages. Furthermore, we give the general system inputs, the assumptions on

participants and the MCS platform, and the system optimization goal to explain in detail the problem formulation.

3.1. Preliminary: Task Allocation Problem in MCS. For most application scenarios, the MCS platform expects enough participant resources to improve the system utility. However, the participants always tend to search for a method that can maximize individual benefits without extra expense. Obviously, the MCS platform's participant requirements and the expected payoff of individuals lead to a contradiction for accomplishing the sensing request efficiently. Unfortunately, such circumstances could be even worse: in practical conditions, the MCS platform with only a limited system budget still needs to respond to the sensing request immediately.

There are two task allocation strategies adopted widely in current works:

(1) Opportunistic Mode. The *opportunistic mode* means that the MCS platform uses a mobility prediction method for participant selection and distributes the sensing requests to the participants with a similar trajectory. Based on the potential participant's historical trajectory data, the MCS platform can adopt machine learning models to predict each participant's daily routine, possible activity, or destinations. After that, the task assigner matches the sensing request with the participant's future trajectory and decides a number of participants as selected workers. The platform that utilizes the *opportunistic mode* for participant profiling can positively affect the delay-unaware sensing requests, the requests with a detailed time schedule, or the recurring sensing requests. In such conditions, the MCS platform can have a longer time to accumulate the participant resource, schedule the sensing request, and respond to the requests properly. While the *opportunistic mode* works efficiently for these sensing requests, it still faces several challenges: first, maintaining a large potential participant pool and accomplishing the profiling task precisely highly rely on the historical data and prediction method; second, for the MCS platform with bursts of sensing requests that occur suddenly, the *opportunistic mode* fails to recruit workers immediately since it needs more time to predict the future trajectory to prepare the participant pool.

(2) Participatory Mode. Instead of utilizing participants' possible trajectory, the selected workers should arrive at the specific locations on time to accomplish the sensing request, which means that they may change their travel plan or daily routine. The participant profiling step can then be regarded as a route planning problem with no historical trajectory requirement. Unfortunately, it also increases the sensing budget since participants may think these changes should be paid more. Besides, this mode could be unreliable, especially when the participants are unwilling to deviate from their original routines or travel plans.

Thus, we expect a task allocation framework that combines the advantages of the existing modes, while having better performance to deal with both the delay-unaware and urgent sensing requests.

3.2. A Semiopportunistic Task Allocation Concept for MCS. Many IoT applications need to make quick reactions to urgent cases, for example, the sensing request of collecting information about a traffic accident or emergency medical care activity. In such circumstances, the sensing requests occur suddenly with a fixed life cycle. The MCS platform can only have very limited time to recruit participants and schedule their sensing resources. Consider the urgent sensing requests' needs and the existing task allocation method's limitations; we proposed our *semiopportunistic mode* by utilizing the "shared mobility" concept. The standard shared mobility service means the shared use of vehicles. For example, a carpooling platform enables shared rides between drivers and passengers with similar origin-destination pairings. The recent success of shared mobility applications indicates that people tend to obtain benefits by making a few changes as possible. Thus, we are inspired to extend this concept into the MCS task allocation framework, which is referred to as the "semiopportunistic" concept in this paper. Under our concept, all potential participants can upload their travel plans with an explicit start point, destination, and deadline based on individual conditions like using carpooling applications, while the MCS platform would regard the travel plans of participants as the fixed constraints. The MCS platform then checks the current sensing requests and provides the participants with routing advice to match their travel plans while maximizing profits. Besides, we consider the system limitations of MCS (like the limited budget or different coverage requirements of task requests) and determine the final task allocation plan.

3.3. Assumptions. In our settings, we consider an MCS system with total budget B, which has a set of sensing requests $\mathrm{TR} = \{t_1, t_2, \cdots, t_n\}$ that occur randomly in a target sensing area L and a set of potential participants $W = \{w_1, w_2, \cdots, w_k\}$ posting their travel plans, while waiting to be paired with the maximized profit path planning advice. We assume target sensing area L is composed of a set of cells, and each cell refers to a possible sensing location. A task request t_i has a target location loc_i, a minimal sensing coverage threshold q_i, a task deadline time_i, and a specific incentive val_i in terms of credits or monetary rewards to encourage participants to respond to this sensing request. Note that the value of q is the minimum coverage required of this sensing request to characterize the target region. When there is less than q_i participants who accept the individual task request, the sensing coverage of t_i will be set to zero. Besides, q of each task request could be different to adapt to each task's needs.

Then, we define a task request as follows:

$$t_i = \langle \mathrm{loc}_i, \mathrm{val}_i, \mathrm{time}_i, q_i \rangle, \quad t_i \in \mathrm{TR}. \tag{1}$$

Furthermore, the travel plan of each potential participant is denoted by a fixed start point sta, a destination des, and also a time constraint time, which indicates that the participant must arrive at the destination location no later than

time. Then, the travel plan can be defined as follows:

$$w_j = \langle \text{sta}_j, \text{des}_j, \text{time}_j \rangle, \quad w_j \in W. \tag{2}$$

The routing advice for participant w_j can be defined as follows:

$$p_{w_j} = \left\{ \text{sta}_j, t_{w_j^1}, \cdots, t_{w_j^m}, \text{des}_j \right\}, \quad p_{w_j} \in P. \tag{3}$$

After the MCS platform provides the routing advice for each potential participant, it checks the system budget and selects a subset of participants W' as the selected workers that could maximize the system utility. Here, we define the system utility as follows:

$$u_{t_i}^{W} = \min \left\{ \left\lfloor \sum_{w_j \in W'} \frac{I_{w_j}^{t_i}}{q_i} \right\rfloor, 1 \right\}, \tag{4}$$

where

$$I_{w_j}^{t_i} = \begin{cases} 0, & \text{when } t_i \text{ not in } p_{w_j}, \\ 1, & \text{when } t_i \text{in } p_{w_j}. \end{cases} \tag{5}$$

Finally, the selected participants would accept the routing advice with a group of sensing task requests inserted in their original trip plan and then sequentially visit the location of each task to collect sensing data.

3.4. Problem Definition. In our proposed task allocation framework based on the semiopportunistic concept, our primary objective is to select a subset of participants with payoff-maximized routes while maximizing MCS platforms' system utility.

Specifically, with the constraint of the given travel plan of the participant, our MCS platform tends to provide the participant w_j with the routing advice p_{w_j} having maximized payoff(p_{w_j}), which can be denoted as follows:

$$p_{w_j}^* = \underset{p_{w_j}}{\text{argmax}} \sum_{t_i \in p_{w_j}} \text{val}_{t_i}. \tag{6}$$

The objective function of system utility is further defined as follows:

$$\begin{aligned} &\max \sum_{t_i \in T} u_{t_i}^{W} \\ &\text{s.t.} \sum_{w_j \in W'} \text{payoff}\left(p_{w_j}\right) \le B, \end{aligned} \tag{7}$$

We can find it is a multiobjective optimization problem when we expect to maximize the above two objectives simultaneously. Unlike the current solutions using Pareto optimality to resolve this problem of having scale limitations, we implement our objective problem in two parts: firstly, for every potential participant, we calculate a route with the maximized payoff to accomplish the participant profiling stage; we further select valid participants from the participant-route set to find the subset with maximized system utility under a limited system budget.

4. Implementation: MCS Task Allocation Framework with Deep Learning Support

This section gives the detailed implementation of our proposed semiopportunistic concept for the MCS task allocation problem, which includes two parts: the participant profiling and the participant selection. First, we give the system overview of our proposed framework. Second, we propose the participant trajectory profiling method based on GNN techniques. We give the detailed GNN model structure and the primary training process to demonstrate the detailed implementation. Finally, an RL-supported participant selection algorithm is proposed. We elaborate on the MDP underlying our method and further explain it by a simplified example.

4.1. Framework Overview. Our proposed working process of the MCS task allocation has three primary stages of fulfilling various types of task requests efficiently, which includes: task request initialization, trajectory profiling for potential participants, and participant selection.

4.1.1. Task Request Initialization. This component accepts the sensing request that randomly occurs. Each request is tagged with a list of the necessary information, including the target location, the maximum requirement of participants, the time of the deadline, and incentive settings. Meanwhile, the platform allows multiple types of sensing requests at one time, which could have different sensing requirements like coverage ratio or time constraints.

4.1.2. Potential Participant Profiling. This component calculates a route with the maximum profits for each participant using the personal travel plan as the available time, start, and destination constraints. Specifically, to motivate the participants to respond to the task requests quickly, the MCS platform calculates the routing advice with the maximum payoff for each participant. Obviously, it is a critical challenge since the routing advice task can be reduced as the orienteering problem [44], which is NP-hard. That means, when the MCS platform has a large set of potential participants and sensing requests, the participant profiling could be the bottleneck. Thus, we introduce an attention-based encoder-decoder model to implement the participant profiling step. Unlike the greedy-based algorithms with approximation bound determined by a fixed utility function, our attention-based model has improved flexibility to adapt to different utility function settings and different sensing area topology scales. It can also adapt to the target sensing area topology changes to adjust to different urban environment dynamics without introducing much extra computation cost.

4.1.3. Participant Selection. Given the participant profiling information, this component determines a group of

participants that can fit the system utility goal under a fixed system budget constraint. In our proposed framework, we give a reinforcement learning-based solution for searching the participant-route pairs. In the typical RL settings, the MCS platform can serve as the RL agent to collect the information of the sensing requirements and participant profiles; then, it actively searches for an optimal subset of participant-route pairs as the selected workers to accomplish the sensing tasks.

4.2. GAT-Based Solution for Potential Participant Profiling. For the sensing requests that randomly occur but expect a quick response, the MCS platform needs to motivate each potential participant efficiently. Thus, in our proposed task allocation framework, the participant trajectory profiling stage serves as an essential part to motivate the participant by providing them with a payoff-maximized route. However, the sensing area could be large. The topology may change due to urban traffic management, accident, or abnormal climate, which indicates that we need a flexible and efficient solution to resolve the topology dynamic challenges while ensuring good scalability.

Specifically, we formalize the routing problem by using a Graph Attention Network to produce embeddings of the target sensing area and then compute the routing advice. As we know, the graph embedding representation can provide a powerful solution to adapt to large-scale urban sensing scenarios. It also supports various types of information that can be represented by the nodes' attributes [45, 46]. Thus, the topology of the target sensing area can be represented as a graph G with a node set $\boldsymbol{\pi} = \{\pi_1, \pi_2, \cdots, \pi_n\}$, where each node could be a potential sensing location. Then, the attention-based encoder-decoder model proposed in [43] is adopted to implement the participant profiling component; we illustrate the working process in Figure 2.

Encoder. The encoder takes each node π_i and its feature val_i in $\boldsymbol{\pi}$ as inputs, and when there is no sensing request at node π_i, the val_i would be 0. The initial node embedding of π_i with parameters W and b can be represented by $h_i^0 = W[\pi_i, \text{val}_i] + b$. By using the N attention layer, h_i^0 can learn the relations with all the other nodes and update itself as h_i^N. Each attention layer has two sublayers, including a multihead attention (MHA) layer and a feedforward (FF) layer. The encoder then uses all the node embedding results to produce the graph embeddings of the target sensing area as follows: $\bar{h}^N = (1/n)\sum_{i=1}^{n} h_i^N$.

Decoder. At time t, the decoder takes the node embedding h_i^N, the sensing area embedding $\bar{h}^N$, and the travel plan of the participant w_j including the start location, destination location, and time constraint $\langle \text{sta}_j, \text{des}_j, \text{time}_j \rangle$ as the inputs. Then, the decoder produces the context embedding $h_{w_j}^N$ of the participant w_j by considering the sensing area embedding $\bar{h}^N$, the location of the previously selected task request at time $t-1$, and the destination des_j, which is denoted as follows:

$$h_{w_j}^N = \begin{cases} \left[\bar{h}^N, h_{\text{sta}_j}^N, h_{\text{des}_j}^N\right], & t = 1, \\ \left[\bar{h}^N, h_{\tau_{w_j}^{t-1}}^N, h_{\text{des}_j}^N\right], & t > 1. \end{cases} \tag{8}$$

After we have $h_{w_j}^N$ and the node embedding of nodes having sensing requests, the decoder uses MHA to get a new embedding result $h_{w_j}^{N+1}$, which indicates the correlation between $h_{w_j}^N$ and other nodes with sensing requests at time t. To produce the routing advice, we use a single head attention layer to determine the next node that the participant w_j should visit. To adapt to the time constraint of w_j, we mask the nodes with a task finishing deadline that the participant w_j cannot visit them within his remaining time. We also mask the nodes that are already visited to ensure the participant would not accomplish a sensing request twice. Such a working process repeats for several iterations until the remaining time runs out. Finally, we can obtain the routing advice for each potential participant. To ensure the final route is payoff-maximized, several training methods can be used to train the encoder-decoder network like the actor-critic algorithm [47] or REINFORCE with deterministic greedy rollout baseline [48].

4.3. Participant Selection with RL Support. Before diving into the details of our proposed RL-based participant selection, we first depict the MDP that formalizes the target problem about selecting a subset of participants to optimize the system utility. Then, we use a tabular Q-learning example to describe the RL working process and further extend it by introducing DNN to ensure our method can adapt to the practical conditions of large-scale participants and sensing requests.

RL is a goal-directed learning approach that uses an interactive manner to explore the environment and investigate how an agent can derive the maximum accumulated reward. To formalize the problem space, the Markov Decision Process is adopted to describe the interactions between the RL agent and the environment, which has three essential components: state, action, and reward. The *state* includes the direct knowledge of the problem to indicate what we have already known at a specific time slice. Then, the RL agent learns how to make *actions* at each state, and it always expects to find the optimal action-state mappings (optimal policy) to maximize the cumulative *reward* as a learning result. Although the problem space may be full of uncertainty, RL can automatically explore the environment and recognize the optimal policy to help the agent always make the right decisions under different conditions. From the above facts, we observe an explicit relation between RL techniques and our participant selection problem, which can be fully explained by the following MDP (Markov Decision Process) formulation.

4.3.1. MDP Formalization. To adopt the RL approach for the participant selection problem, we regard the MCS system as the environment. The RL agent interacts with the

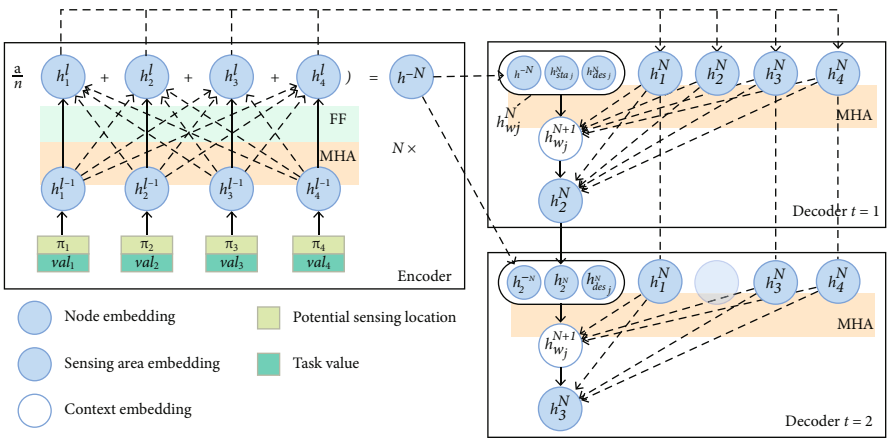


Figure 2: Model structure and working process.

environment to gather information about the available participant profiles (planned routes), current task requests, and system budget requirements. Through the RL agent's exploration and exploitation, the agent learns the optimal policy to decide the participant-task mappings to achieve better system utility. To implement the above process, we first define the state, action, and reward as three primary elements and depict our MDP in Figure 3.

State. We represent the functional conditions of the MCS platform—the current task request TR, system budget B, available participant resources, and related profiles P—as the state of our MDP. Thus, the state s_t at time t can be represented as $s_t = (\mathrm{tr}_t, b_t, p_t)$ that belongs to the state space S.

Action. We assume there are k participant profiles in the current MCS platform, and at each time step, the RL agent selects one participant to accomplish the tasks in his route. Then, the action space is given by $A = \{1, 2, \cdots, k\}$ and action a_t represents the decision on participant selection. With each available action, the RL agent observes a state transition. When a participant's route is selected to improve the system utility, the next state will be updated accordingly.

Reward. The reward signal guides the agent towards an optimal solution for the target problem. In our problem settings, the objective is maximizing the total system utility under resource constraints. Specifically, we set the reward at each time step as the system utility. Note that although the agent can receive an immediate reward for each action participant selection, the RL agent focuses more on maximizing the cumulative reward to ensure the MCS system receives the largest utility value by the selected participant group. To ensure the RL agent can be farsighted, we use the discount rate $\gamma \in [0, 1]$ as the parameter to determine the present value of the future reward. As its value approaches 1, it means that the RL agent takes future rewards into account more strongly.

4.3.2. A Tabular Q-Learning Example. For small-scale applications, the MCS system can use tabular Q-learning, a value-based reinforcement learning algorithm that uses an evaluation concept—Q-function—to derive the optimal policy. In the tabular Q-learning-based participant selection algorithm, it utilizes the Q-function denoted as $Q(s, a)$ to calculate the maximum expected future reward (system utility) that the agent will get if it takes action a at state s. Afterward, each possible state-action pair's Q-value will be stored in the Q-table $Q_{|S|\times|A|}$. Thus, the RL agent can evaluate each participant selection in terms of reward, derive the estimated value of $Q(s, a)$, and record this value in $Q_{|S|\times|A|}$. To find the optimal policy, the Q-table $Q_{|S|\times|A|}$ will be further iterated and updated by the Bellman equation with learning rate α as follows:

$$Q(s, a) = Q(s, a) + \alpha\left(r + \gamma \max_a Q\left(s', a\right) - Q(s, a)\right). \quad (9)$$

After $Q_{|S|\times|A|}$ is updated, the available action space will be changed accordingly so that the agent can select another participant satisfying the current budget constraints. Through the tabular Q-learning process, the agent follows the ε-greedy policy, that is, with $1 - \varepsilon$ possibility to select the participants with the largest Q-value until the Q-table $Q_{|S|\times|A|}$ is converged. That means, under each state s, the agent can select the participant with the largest Q-value by searching the Q-table $Q_{|S|\times|A|}$ as an optimal policy.

To further explain the training process, we introduce a simplified participant selection problem as a toy example illustrated in Figure 4.

In our example, we assume that the MCS system has 10 task requests with different participant requirements denoted as task$_i d : q$; for example, tr0 : 2 indicates that task 0 has a minimum requirement of 2 participants. We also have 4 participants with profiles (planned routes) denoted as Participant A (tr0 – tr1 – tr2), Participant B (tr1 – tr4 – tr7), Participant C (tr3 – tr6 – tr7), and Participant D (tr7 – tr8). For the tabular Q-learning process, we set the discount factor γ and learning rate α as 1. First, at state s_0, the MCS system has not chosen any participants, and the Q-table is initialized with 0. Then, the RL agent begins to interact with the environment by a random policy, and we assume it chooses action a_2 that selects *Participant B*. The RL agent receives an immediate reward in terms of current system utility, and we observe a state transition from s_0 to s_1. We can

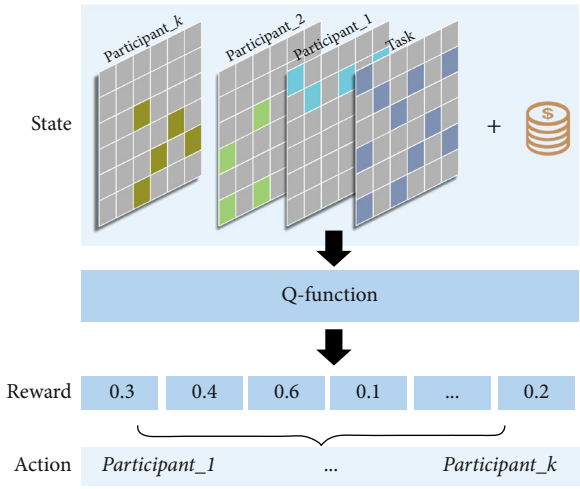


Figure 3: Illustration of the MDP.

now update the $Q(s_0, a_2)$ in the Q-table as $Q(s_0, a_2) = 0.2 + 0 = 0.2$.

Similarly, at the next time step, we use the simple greedy policy and choose action a_4 that selects *Participant D* to update $Q(s_1, a_4)$ as $Q(s_1, a_4) = 0.3 + 0 = 0.3$. After several rounds, at time step t_k, the Q-table is changed as in the figure: t_k. At this time step, we assume the RL agent goes back to s_0 and it checks the current Q-table and recognizes that under s_0, $Q(s_0, a_2)$ has the largest Q-value. Hence, the RL agent executes action a_2 and observes a state transition from s_0 to s_1. According to the Q-table at time step t_k, the largest expected future reward is 0.3. Hence, at the next time step t_{k+1}, $Q(s_0, a_2)$ will be updated as $Q(s_0, a_2) = 0.2 + 0.3 = 0.5$. That means, when the RL agent meets s_0 at future time steps, it has a higher possibility to select action a_2, since the Q-table has explicit evidence that it can lead to higher accumulating rewards. As an iterative process, the above operations repeat several times and the Q-table will be improved at each iteration so that the Q-value is approaching the practical state-action value.

4.3.3. DDQN-Based Participant Selection Algorithm. Although tabular Q-learning offers an effective solution for our simplified participant selection problem, we still require a method meeting the practical requirements, such as large-scale participants, system budget constraints, and various attributes of task requests. Thus, instead of using the Q-table in the tabular Q-learning method, we choose the Convolution Neural Network (CNN) as a Q-network to obtain the estimation of the Q-function. Specifically, we represent the state of our MDP—the current allocation task request waiting to be scheduled, available participant resource, and planned route profiles—as a $m \times n \times (i + j)$ matrix. Here, $m \times n$ indicates the target map has $m \times n$ cells, i is the number of participants, and j is the number of task request' attributes. With two convolutional layers and one fully connected layer, our proposed CNN is used to extract the above state matrix's primary feature and output the Q-function value for each state. We further illustrate the Q-network training via the DDQN-based algorithm in Algorithm 1 as follows:

In the DDQN-based participant selection algorithm, we utilize the experience replay technique to break the temporal correlations that lie in various training episodes. A replay buffer with a fixed size is utilized to mix experiences at different time steps for the Q-network updates. At the beginning of this algorithm, the Q-network is initialized to a random value (Line 3). Meanwhile, the initial state s_0 feeds the Q-network and the RL agent selects an action under the ε-greedy policy to start the first training episode (Lines 5 and 6). Next, the state transition $\{s, a, r, s'\}$ is stored in the replay buffer (Line 8). When executing an action, the algorithm checks the available system budget to ensure the remaining budget can afford the next participant selection (Line 9). Then, given the replay buffer, the agent samples a random minibatch and updates the Q-network using the following loss function (Lines 14 and 15):

$$\text{Loss}(\theta) = \frac{1}{m} \sum_{\text{minibatch}} \left[\left(r + \gamma \max_{a'} Q_{\theta'}(s', a') - Q_{\theta}(s, a) \right)^2 \right]. \tag{10}$$

Here, the target Q-network is an independent estimator that updated slower than the Q-network, to avoid maximization bias by disentangling updates from biased estimate values.

5. Performance Evaluation

This section validates our proposed method through extensive simulations of multiple application scenarios. We first introduce the experimental setup, parameter settings, and baseline algorithms. Then, we demonstrate the performance comparison result in multiple scenarios having different system budgets, numbers of participants, or numbers of sensing requests.

5.1. Dataset and Selected Parameters. For generating the trip plans of the potential participants, we adopt the T-drive dataset [49] to provide the start location and destination, which contains the GPS trajectories of 10,357 taxis from Feb 2nd to Feb 8th, 2008. We then select 1000 travel plans and randomly generate a deadline as the time constraint of each trip plan to form our potential participant pool. Furthermore, since we propose a two-stage solution to implement the semi-opportunistic MCS task assignment concept we defined in this paper, four primary factors that affect the simulation results are selected to describe the validity and performance: the total system budget of the MCS platform, the number of task requests, the value of the task, and the number of participants.

5.2. Experimental Setup and Settings. We implement our work on the PyTorch platform. The encoder-decoder model in [19] is adopted to provide the route with a maximized payoff. The minimum requirement of participants for each task request is randomly generated from $(2, 15)$, while the value

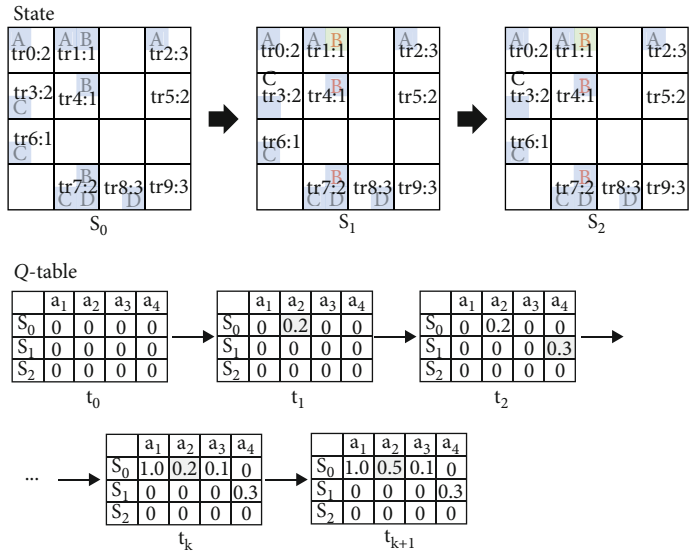


Figure 4: A tabular Q-learning example.

Participant selection based on DDQN.
1: Initialize Q-network, Q' target network with θ
2: **while** In each episode **do**
3: Initialize s
4: **for** step in episode **do**
5: With probability ε, select a random action a
6: otherwise select $a = \max_a Q_\theta(s; a)$
7: $r, s' = \text{perform\_action}(s, a)$
8: store transition $\{s, a, r, s'\}$ in replay_buffer
9: **if** current system budget cannot afford any participant **then**
10: s is terminal state
11: **else**
12: $s = s'$
13: **end if**
14: sample random minibatch(m) of transitions from replay_buffer
15: perform minibatch gradient descent
16: every updated period $T, \theta' = \theta$
17: **end for**
18: **end while**

Algorithm 1:

of the task request is randomly generated from $(10, 50)$. For clarity, we take the routing advice for 15 participants as an example shown in Figure 5. The sample routes are the example solutions for a sensing area consisting of 100×100 cells and 200 sensing requests using the GAT-supported routing method. For the proposed RL-based task allocation solution using the DDQN algorithm, the Q-network is CNN-based. The replay buffer size is 1000, and the minibatch size for sampling is 32. We set the learning rates of the Q-network as 10^{-3} and the discounting factor γ as 0.99.

5.3. Baseline Algorithm. Since no previous works have studied the task allocation method with the semiopportunistic concept via deep learning support, we select the following baseline methods to accomplish the comparative studies:

Random allocation. This method randomly selects participants from the potential participant pool until meeting the total system budget. Since the random character may affect the simulation result, in our simulation, we repeat this method for 10 times and the average overall utility is utilized as the final result.

Low-payoff first allocation. This is a single-loop greedy algorithm which tends to select more participants to obtain higher system utility. It orders the potential participants from the minimal total payoff and then selects a subgroup of participants having a route with a low total payoff until the total system budget runs out.

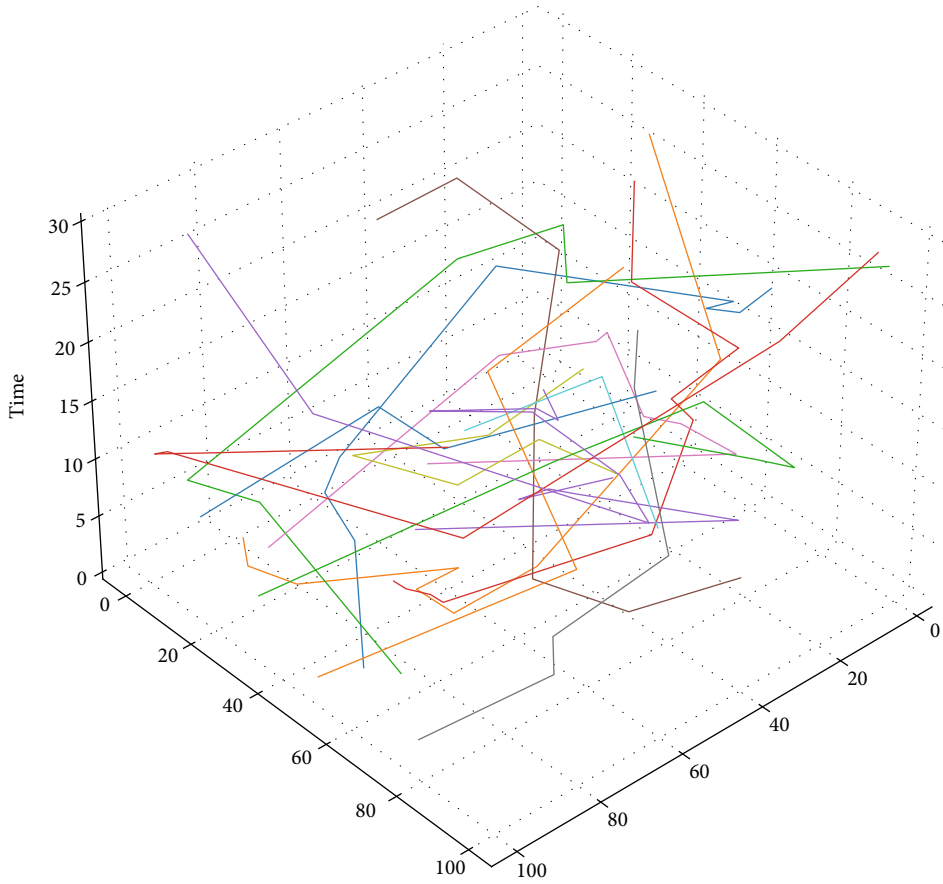


Figure 5: Example routing advice with a maximized payoff.

Long-route first allocation. This is a single-loop greedy algorithm which tends to select participants with the long-route first method to reach the system utility goal, since the longer route means this participant could obtain more sensing request than others. It orders the participant by the route length and then selects a subgroup of participants with the longer route until the total system budget runs out.

5.4. Performance Comparison via Multiple Scenarios

5.4.1. Different Numbers of Task Requests. Figure 6 depicts the performance comparison result on the system utility among the DDQN-based allocation and other baseline methods under the different numbers of task requests, where we fix the number of selected participants as 200. We observe that in the beginning, the performance difference on the system utility is not very significant due to the very small task numbers. When the system utility decreases with the increasing number of tasks for all three methods due to the limited resource's enhanced competition, our proposed algorithm outperforms other baseline methods to obtain higher utility for different settings for the number of task requests. When the number of requests increases while all the three methods' utility decreases, our method's system utility still outperforms the other methods, and the utility decreases more steadily. Although we can observe the same decreasing trend with the other three methods, they fail to obtain a similar utility performance as our method does. Such results indicate that our method works well at both the small- and large-scale sensing requests.

5.4.2. Different Values of the Total System Budget. Figure 7 gives the results for the system utility changes among all the four allocation methods when the system budget is varied. In this experiment, we define the system budget as $B = \sum_{i \in \mathrm{TR}} \mathrm{val}_i \times q_i$. When the total budget increases, it indicates that the MCS platform can have more participants to accomplish the request. Therefore, we observe an increase of all four methods. At the beginning, except for the utility of the random method that is dragging by the random character, all the other three methods have a similar increasing trend. Then, the performance gap between our method and the other three methods becomes larger. From Figure 7, we find the random method has the smallest utility increment while the low-payoff first method and the long-route first method have a very similar trend in the end. At the same time, our method is more stable and can always obtain higher utility when the system budget changes.

5.4.3. Performance Comparison for Different Values of q of Each Task Request. Figure 8 shows the comparison result when we are varying the values of q for each task request. We generate q, the minimum requirement of the participant for each request, by randomly choosing a number from (5,

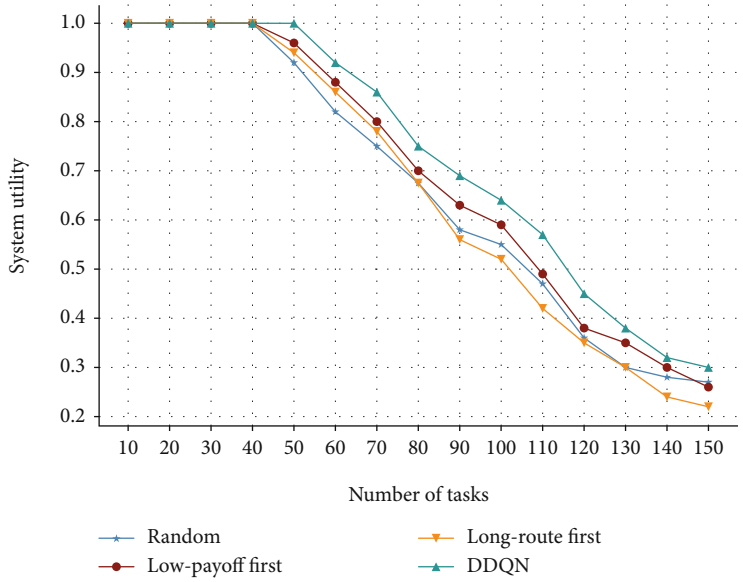


Figure 6: Performance comparison for the various numbers of tasks.

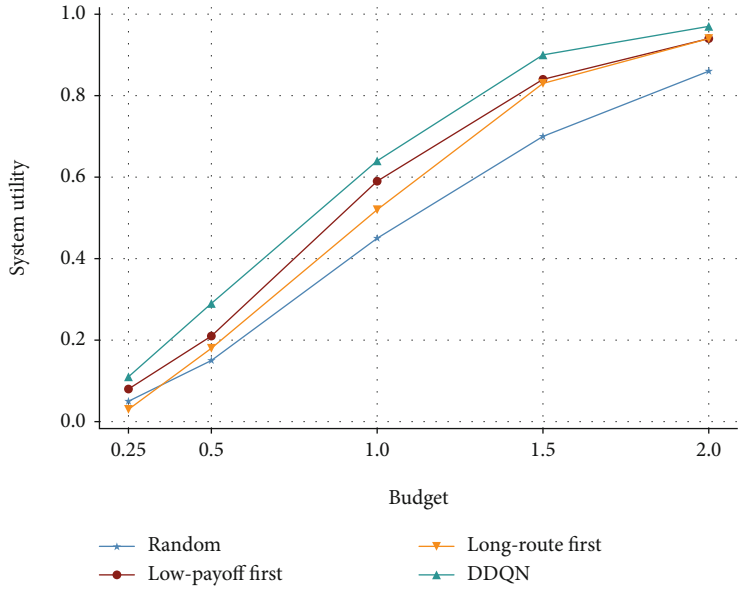


Figure 7: Performance comparison for the different values of the budget.

15). Figure 8 plots the utility changes when the value of q varies, which can simulate the multiple types of the minimum thresholds for different application scenarios. When the requirement of the task request increases, we observe that the system utility decreased since the total budget and potential participant pool are fixed. The system utility of the random method decreased sharply, while the low-payoff first method and the long-route first method represent similar trends with our method; however, when the value of q increases from 8, we observe that both of them have a noticeable decline. Thus, compared to other baseline methods, our method can obtain a significantly high system utility that decreases more steadily during the value changes of$q$.

5.4.4. Number of Assigned Participants for Each Task Request. Figure 9 represents the changes in the number of assigned participants when varying the total number of task requests. In this experiment, we assume all the 100 task requests have the same minimum threshold $q = 5$, which is represented by the dotted line in Figure 9. We can see that for each task request, the long-route first method tends to select the participant with the longer route to obtain the system utility goal; however, it could waste several participants. Meanwhile, we observe a similar result of the low-payoff first method. For example, we can notice for task id:40 that it allocates 7 more participants, which is significantly larger than the minimum threshold. Compared with the three baseline models, our

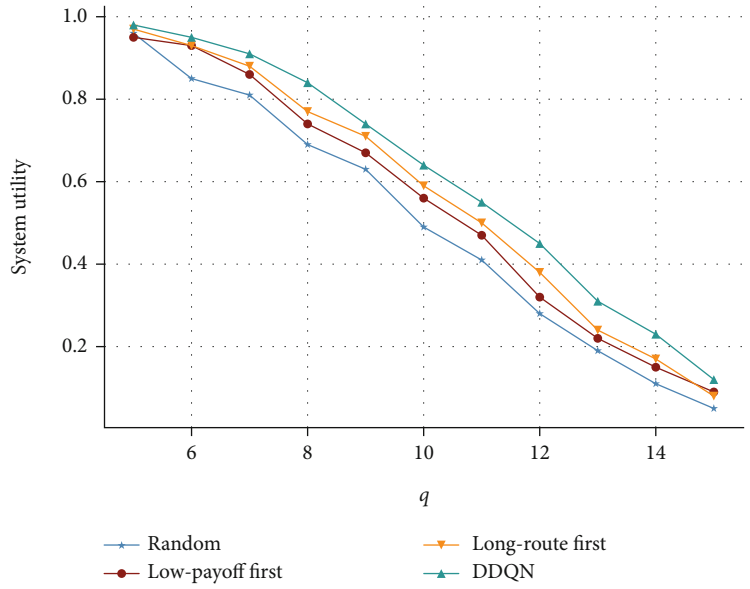


Figure 8: Different values of q for each task request.

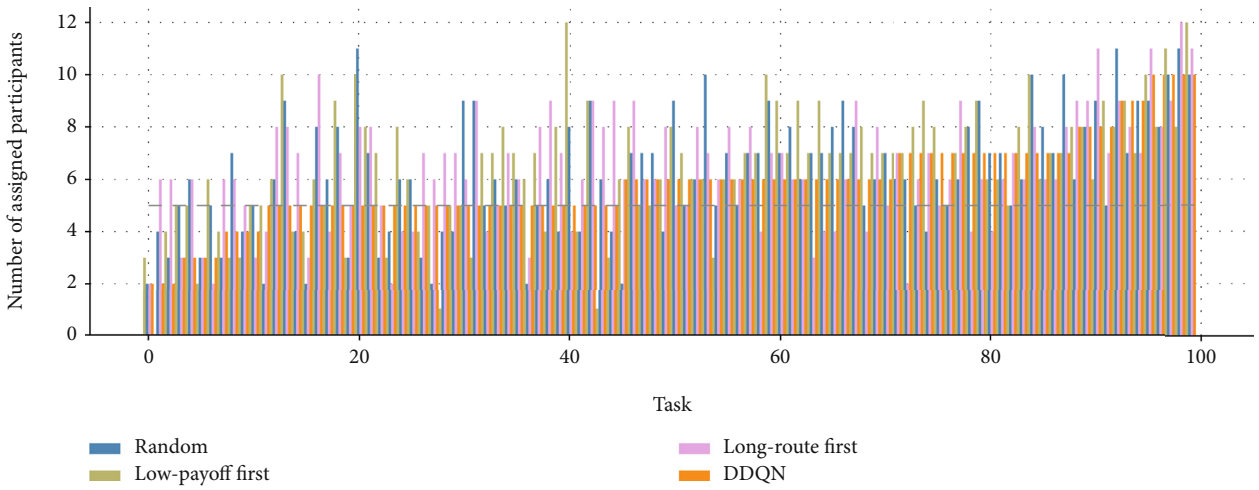


Figure 9: Number of assigned participants for each task request.

method tends to meet the threshold with less waste of participant resources so that we can see that the number of assigned participants of all the 100 task requests is closer to the dotted line.

6. Discussion and Conclusion

This paper studied the task allocation problem in MCS systems by introducing a novel semiopportunistic concept inspired by "shared mobility" applications. Meanwhile, we aim to maximize the payoff of the participant by producing a well-considered route. Then, we use the reinforcement learning technique to select a subgroup of participants under the system utility optimization goal. Our proposed solution has several advantages. First, to implement our proposed framework efficiently, we adopt a representation learning approach to produce the target sensing area embeddings. At the same time, output a payoff-maximized route for each participant. Second, the reinforcement learning-based participant selection algorithm is proposed for selecting a subgroup of participants that can meet the system utility goal. Unlike traditional solutions using greedy-based or heuristic-based algorithms, our proposed framework and its implementation can support large-scale sensing requests and various types of utility optimization goals. Finally, extensive simulations indicate our solution outperforms the baseline methods under various conditions. In the future, we plan to extend our method into the social network-based MCS platforms to investigate the participant profiling problem with social influence analysis. We expect that investigating social relationships among participants by using deep learning approaches could be a different solution to solve the participant resource bottleneck.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key Research and Development Plan of China under Grant 2017YFA0604500, in part by the National Natural Science Foundation of China under Grant 61701190, in part by the Youth Science Foundation of Jilin Province of China under Grant 20180520021JH, in part by the Youth Sci-Tech Innovation Leader and the Team Project of Jilin Province of China under Grant 20170519017JH, in part by the Key Technology Innovation Cooperation Project of Government and University for the whole Industry Demonstration under Grant SXGJSF2017-4, and in part by the Program for Innovative Postdoctoral Talents in Shandong Province.

References

[1] D. Wu, T. Xiao, X. Liao et al., "When sharing economy meets IoT: towards fine-grained urban air quality monitoring through mobile crowdsensing on bike-share system," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–26, 2020.

[2] X. Wang, Z. Ning, X. Hu et al., "A city-wide real-time traffic management system: enabling crowdsensing in social internet of vehicles," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 19–25, 2018.

[3] M. Mehdi, G. Mühlmeier, K. Agrawal, R. Pryss, M. Reichert, and F. J. Hauck, "Referenceable mobile crowdsensing architecture: a healthcare use case," *Procedia Computer Science*, vol. 134, pp. 445–451, 2018.

[4] Y. Wang, Z. Cai, Z.-H. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1033–1046, 2020.

[5] H. Yang, F. Li, D. Yu, Y. Zou, and J. Yu, "Reliable data storage in heterogeneous wireless sensor networks by jointly optimizing routing and storage node deployment," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 230–238, 2021.

[6] Z. Duan, W. Li, X. Zheng, and Z. Cai, "Mutual-preference driven truthful auction mechanism in mobile crowdsensing," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1233–1242, Dallas, TX, USA, October 2019.

[7] S. Reddy, D. Estrin, and M. B. Srivastava, "Recruitment framework for participatory sensing data collections," in *Pervasive Computing. Pervasive 2010*, P. Floréen, A. Krüger, and M. Spasojevic, Eds., vol. 6030 of Lecture Notes in Computer Science, , pp. 138–155, Springer, 2010.

[8] Y. Liu, L. Kong, and G. Chen, "Data-oriented mobile crowdsensing: a comprehensive survey," *IEEE Communication Surveys and Tutorials*, vol. 21, no. 3, pp. 2849–2885, 2019.

[9] J. Li, H. Jiao, J. Wang, Z. Liu, and J. Wu, "Online real-time trajectory analysis based on adaptive time interval clustering algorithm," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 131–142, 2020.

[10] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[11] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: challenges, solutions, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2419–2465, 2019.

[12] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[13] R. Tachet, O. Sagarra, P. Santi et al., "Scaling law of urban ride sharing," *Scientific Reports*, vol. 7, no. 1, article 42868, 2017.

[14] G. Laporte, F. Meunier, and R. W. Calvo, "Shared mobility systems: an updated survey," *Annals of Operations Research*, vol. 271, no. 1, pp. 105–126, 2018.

[15] N. Chan and S. Shaheen, "Ridesharing in North America: past, present, and future," *Transport Reviews*, vol. 32, pp. 112–193, 2012.

[16] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, July 2019.

[17] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[18] X. Fan, X. He, C. Xiang et al., "Towards system implementation and data analysis for crowdsensing based outdoor RSS maps," *IEEE Access*, vol. 6, pp. 47535–47545, 2018.

[19] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *35th Annual IEEE International Conference on Computer Communications, INFOCOM*, pp. 1–9, San Francisco, CA, USA, April 2016.

[20] Z. Cai, Z. Duan, and W. Li, "Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[21] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 635–644, Atlanta, GA, USA, June 2017.

[22] D. Zhang, H. Xiong, L. Wang, and G. Chen, "CrowdRecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *UbiComp '14: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 703–714, Seattle, WA, USA, September 2014.

[23] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: a framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 392–403, 2017.

[24] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2016.

[25] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "iCrowd: near-optimal task allocation for piggyback crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, 2016.

[26] J. Wang, F. Wang, Y. Wang, D. Zhang, L. Wang, and Z. Qiu, "Social-network-assisted worker recruitment in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1661–1673, 2019.

[27] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Computer Networks*, vol. 102, pp. 157–171, 2016.

[28] T. Luo, S. S. Kanhere, J. Huang, S. K. Das, and F. Wu, "Sustainable incentives for mobile crowdsensing: auctions, lotteries, and trust and reputation systems," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 68–74, 2017.

[29] J. Li, Z. Cai, J. Wang, M. Han, and Y. Li, "Truthful incentive mechanisms for geographical position conflicting mobile crowdsensing systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 324–334, 2018.

[30] Z. Wang, C. Wang, X. Ye, J. Pei, and B. Li, "Propagation history ranking in social networks: a causality-based approach," *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 161–179, 2020.

[31] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[32] G. Yang, Y. Li, Y. Song et al., "A real-time recommendation algorithm for task allocation in mobile crowd sensing," in *Wireless Algorithms, Systems, and Applications - 15th International Conference, WASA 2020, Proceedings, Part I*, vol. 12384 of *Lecture Notes in Computer Science*, pp. 640–652, Qingdao, China, September 2020Springer.

[33] S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, "zkCrowd: a hybrid blockchain-based crowdsourcing platform," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4196–4205, 2020.

[34] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, 2016.

[35] S. Arora, "A survey on graph neural networks for knowledge graph completion," 2020 https://arxiv.org/abs/2007.12374.

[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, https://arxiv.org/abs/1710.10903.

[37] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: a graph multi-attention network for traffic prediction," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence 30 Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, pp. 1234–1241, New York, NY, USA, February 2020, AAAI Press.

[38] S. Wu, W. Zhang, F. Sun, and B. Cui, "Graph neural networks in recommender systems: a survey," 2020, https://arxiv.org/abs/2011.02260.

[39] Z. Zhang, F. Zhuang, H. Zhu, Z. Shi, H. Xiong, and Q. He, "Relational graph neural network with hierarchical attention for knowledge graph completion," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020,*, pp. 9612–9619, New York, NY, USA, February 2020, AAAI Press.

[40] R. S. Sutton and A. G. Barto, *Reinforcement Learning - An Introduction. Adaptive Computation and Machine Learning*, MIT Press, 1998.

[41] Y. Dai, D. Xu, Y. Lu, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for edge caching and content delivery in internet of vehicles," in *2019 IEEE/CIC International Conference on Communications in China, ICCC 2019*, pp. 134–139, Changchun, China, August 2019, IEEE.

[42] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[43] W. Kool, H. van Hoof, and M. Welling, "Attention, learn to solve routing problems!," in *7th International Conference on Learning Representations, ICLR 2019*, New Orleans, LA, USA, May 2019.

[44] C. Chekuri, N. Korula, and M. Pál, "Improved algorithms for orienteering and related problems," *ACM Transactions on Algorithms*, vol. 8, no. 3, pp. 1–27, 2012.

[45] K. Yang, J. Zhu, and X. Guo, "POI neural-rec model via graph embedding representation," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 208–218, 2021.

[46] Z. Ye, H. Zhao, K. Zhang, Z. Wang, and Y. Zhu, "Network representation based on the joint learning of three feature views," *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 248–260, 2019.

[47] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," in *5th International Conference on Learning Representations, ICLR 2017, Workshop Track Proceedings*, Toulon, France, April 2017.

[48] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, pp. 229–256, 1992.

[49] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *KDD '11: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 316–324, San Diego, CA, USA, August 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6691975, 12 pages
https://doi.org/10.1155/2021/6691975

Research Article

Protecting the Moving User's Locations by Combining Differential Privacy and k-Anonymity under Temporal Correlations in Wireless Networks

Weiqi Zhang,[1] **Guisheng Yin**,[1] **Yuhai Sha**,[1] **and Jishen Yang**[2]

[1]*College of Computer Science and Technology, Harbin Engineering University, Heilongjiang, China*
[2]*Department of Computer Science, Georgia State University, Georgia, USA*

Correspondence should be addressed to Weiqi Zhang; zhangweiqi@hrbeu.edu.cn

Received 11 November 2020; Revised 21 December 2020; Accepted 12 January 2021; Published 2 February 2021

Academic Editor: Xiao Zhang

The rapid development of the Global Positioning System (GPS) devices and location-based services (LBSs) facilitates the collection of huge amounts of personal information for the untrusted/unknown LBS providers. This phenomenon raises serious privacy concerns. However, most of the existing solutions aim at locating interference in the static scenes or in a single timestamp without considering the correlation between location transfer and time of moving users. In this way, the solutions are vulnerable to various inference attacks. Traditional privacy protection methods rely on trusted third-party service providers, but in reality, we are not sure whether the third party is trustable. In this paper, we propose a systematic solution to preserve location information. The protection provides a rigorous privacy guarantee without the assumption of the credibility of the third parties. The user's historical trajectory information is used as the basis of the hidden Markov model prediction, and the user's possible prospective location is used as the model output result to protect the user's trajectory privacy. To formalize the privacy-protecting guarantee, we propose a new definition, L&A-location region, based on k-anonymity and differential privacy. Based on the proposed privacy definition, we design a novel mechanism to provide a privacy protection guarantee for the users' identity trajectory. We simulate the proposed mechanism based on a dataset collected in real practice. The result of the simulation shows that the proposed algorithm can provide privacy protection to a high standard.

1. Introduction

In recent years, the booming amount of personal mobile devices with location services has promoted the development of location-based systems in wireless networks [1]. The widespread use of mobile smart devices has laid the foundation for massive data collection based on mobile perception. In these data collection systems, location-based services (LBSs) provide real-time services related to the user's current location information. Various useful applications depend on LBSs. For example, Google Maps provides navigation services such as route suggestions and road traffic condition notifications. Groupon and Yelp [2] provide business service information based on the distances from the users' location. Although LBSs (as shown in Figure 1) are very useful and convenient for users, these conveniences are at the expense of users' private information. The service providers can infer an individual's residence information, work location, and other private information by observing the user's temporal correlated data [3–6].

Many methods for personal information protection are proposed as follows. One of the solutions is Private Information Retrieval (PIR) [7]. In modern cryptography, the main purpose of PIR is to allow a user to retrieve items from a server without disclosing any private information. In other words, the server does not know the user's specific query information and retrieved data in the process. However, one major disadvantage of this technique is the enormous amount of calculation for redesigning different queries according to different query types. Most of the methods are

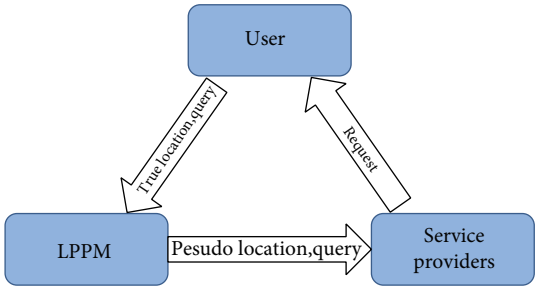


Figure 1: Location-based services.

developed based on location obfuscation, which uses a cloaking area or a perturbed location. These solutions rely on syntactic privacy models, which cannot provide a strict privacy guarantee. Unfortunately, most of the solutions only consider the stationary scenario and perturb the location at a single timestamp while neglecting the temporal correlation of the movement of the user's locations. Hence, the adversaries can effortlessly access more private information by linking inference attacks. Most typical methods to protect users' private information use differential privacy and k-anonymity. k-Anonymity as one of the principal approaches [8–10] ensures the probability of success to any linking attack to be lower than $1/k$. However, it provides a lower privacy guarantee and data utility.

Differential privacy [11] was originally proposed by Dwork in 2006. Later, the idea is regarded as a standard for private information preservation. Although the applications of differential privacy in protecting private information have gradually become applicable in practice, some challenges still exist in the problem of continuous location sharing. First of all, in the standard privacy protection settings, only user-level privacy (whether a user appears in a dataset or not) is protected. In our setting, the trajectories of a single user are protected for a period of time. Second, the released trajectory can be identified based on road networks without temporal correlation. Furthermore, the adversary can identify the user captured by moving patterns. Finally, none of the effective released trajectory mechanism utilizes the combination of k-anonymity and differential privacy.

In this paper, we propose an all-new solution to preserve the user's trajectory privacy with k-anonymity and differential privacy. As shown in Figure 2, a moving user needs to continuously share locations with untrusted service providers or other third parties in a period of time. In other words, in our solution, a user's accurate location information is only known by him. We regard all service providers and third parties as adversaries. The adversaries have side knowledge as much as they can obtain. We propose a new privacy protection system that enables private location sharing without disclosing users' accurate locations to these adversaries and protects users' trajectories in a continuous time period.

The proposed system is noted as UGIS (User and Geographic Space-Indistinguishable System), and this system consists of two parts. One part is the KD-location region (KD is the k-anonymity and differential privacy for short), referred to as a special region in the context. Another part

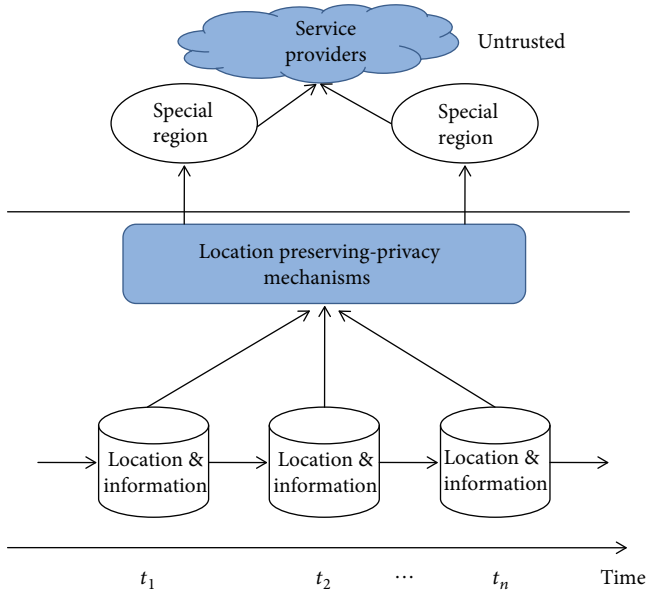


Figure 2: Processing mechanism.

is the users' accurate trajectory processing mechanism. In the KD-location region, adversaries cannot recognize the target user. We then move on to the trajectory processing mechanism that makes a good performance to protect users' trajectories. To our knowledge, UGIS is a better private processing mechanism that combines k-anonymity and differential privacy methods to protect the location and trajectory information of users. The following is a summary of this paper's contribution:

(i) To protect the user's accurate location, we only need to "hide" it in a special region set in which the adversaries cannot distinguish the locations or users. Accordingly, we propose a special region set based on k-anonymity and differential privacy to protect the accurate location of each timestamp

(ii) To show that the user's movement is associative and temporally correlated. In our problem, the user's location transfer is time-related. We use the Markov model to represent the user's location change between continuous timestamps [12]. Hence, from the perspective of adversaries, the user's location transfer model is a hidden Markov model (HMM)

(iii) We focus on the transfer mechanism in a Markov model. We utilize the concept of differential privacy to add noise to this transfer mechanism to make the users' trajectories indistinguishable

The rest of this paper is organized as follows. Section 2 is our related works. In Section 3, we discuss the notions of location privacy from the literature and analyze the weaknesses and strengths of the state-of-the-art algorithms. Also, in Section 3, we introduce the coordinate system and location transition model. We then describe several components and definitions of our UG-indistinguishable system in Section 4.

Section 5 contains the illustration of the framework and the implementation of our location release algorithm. The experiment and evaluation are presented in Section 6.

2. Related Works

In this part, we mainly make some generalizations and summaries of the previous literature. A few recent works [3–15] provide an overview of location privacy protection mechanisms (LPPMs) and methods. These location privacy protection mechanisms mainly use obfuscation technology to achieve the anonymity area. The most widely used approach to construct the LPPMs is k-anonymity. The notion of k -anonymity is commonly used to protect privacy for location-based systems in most of the works in the field. These systems mainly focus on protecting the users' identities and preventing the adversary from inferring accurate information among k users from the published user datasets. One way to implement this method is to use dummy locations mentioned in [16, 17]. However, since the output of the dummy location is controlled by the server, the adversaries can easily find out where the dummy location is not logically generated. Another method to achieve k-anonymity is through the cloaking region [18–20]. The disadvantage of the method is the high risk of having a too-large cloaking area to satisfy the$k$value in the scene with few users. A different approach is to add certain quality constraints to provide better privacy protection [21], while [22] additionally using bandwidth constraints. Literature [7] also proposed a location privacy mechanism focusing on the evaluation based on location-based range queries. This method evaluates the degree of privacy according to the size of the cloaking area and the coverage of the sensitive area. Two methods have been proposed to deal with the adversary's background knowledge, by expanding the anonymous area or delaying the sending of requests. Both solutions may lead to a decline in service quality. The methods based on k-anonymity are improved, but the definition of differential privacy provides a more rigorous guarantee.

Several privacy-protecting methods use the differential privacy approach in recent works [23, 24]. For instance, [25] presents a way to statistically simulate the location data from a database while providing privacy guarantees. They designed an information perturbation mechanism to generate aggregated information from a large amount of locations, trajectories, and spatiotemporal data [26–29]. [30] proposed a differential privacy data mining algorithm that uses a spatial quadtree decomposition technique to preprocess the locations. The work closest to ours is [31]. A large part of the research is based on the use of cloaking areas to enforce location confusion mechanisms. This method leads to a reduction in the utility of published data. [32] proposed a data sanitization method collectively manipulating users' profiles and friend relationships. This method is not suitable for our framework setting and further research. However, the method does not solve the users' movement trajectory problem. In this paper, our system protects the users' accurate location with a rigorous privacy guarantee and makes the users' trajectories indistinguishable at each timestamp.

3. Preliminary

In this section, we discuss various notions of location privacy-preserving methods such as k-anonymity, differential privacy, and location transfer model. We consider a scenario where a user wants to post a query about points of interest at the current location by using a personal device (e.g., smartphone) to query a public service provider. The users expect their accurate location to be private regardless of the process of the search. Our goal is to develop a real-time privacy mechanism that provides privacy protection in a formal notion to achieve users' expected privacy protection level. A list of frequently used symbols in this paper is all motioned in Table 1.

3.1. k-Anonymity. k-Anonymity is one of the privacy protection methods widely used in most location-based systems. These systems focus on protecting the user's identity, making the adversary unable to infer which user is the true target among k users. One way is to generate $k-1$ properly pseudo points and use the actual location and pseudo locations to perform k queries to the service provider. Another way to achieve k-anonymity is through a cloaking area. This approach involves creating a cloaking area that includes k users sharing some points of interest, then querying the server using this cloaking area instead of the accurate location. Unfortunately, the adversaries can identify the target user when adequate side knowledge is available. Pseudo locations are only useful if they have enough similarity with the real locations from the adversaries' point of view.

As a result, notions that abstract from adversaries' knowledge, such as differential privacy, have more popularity later than k-anonymity approaches.

3.2. Differential Privacy and Laplace Mechanism. Differential privacy (DP) [11] is a notion of private information inspired by the concept of statistics. DP guarantees to maximize the accuracy of data queries when querying from the statistical database while minimizing the chance of identifying other records. DP removes the individual characteristics while preserving statistical characteristics to protect the user's privacy. DP has gradually become the de facto standard in data privacy due to its strong privacy guarantees in statistical analysis. Moreover, differential privacy is a semantic model that does not need to rely on the adversary's background knowledge and provides a higher level of semantic security from private information. Differential privacy ensures that adversaries cannot infer whether a particular user is present in the original data. Releasing data according to differential privacy ensures that adversaries cannot infer any information about personal information from the "sanitized data." The definition of differential privacy is demonstrated as follows.

Table 1: Summary of notations.

Symbol	Meaning
a_u^t	User's information at timestamps t
X_i, Y_i	Axis in the coordinate system
Δ	The special location set a user is located in
$p_t[i]$	The probability of a user in region i at timestamps t
l	Current location of the user
l'	Puppet of the user's location
R_t	The special region a user is located in
$\widehat{l}$	Nearest location to the target user's location in the special region
S_f	The sensitivity
TM	Markov transfer mechanism
D, D'	Neighboring database
T_{a_u}	Trajectory of a user

Definition 1. (differential privacy). A mechanism M satisfies ε-differential privacy if any outputs $\in S$ and database D and its neighboring database D' can be obtained by either adding or removing a single record, and the following holds:

$$\Pr\left(M(D) \in s\right) \leq e^{\varepsilon} \times \Pr\left(M\left(D'\right) \in s\right). \tag{1}$$

The Laplace mechanism [33] is commonly used to achieve ε-differential privacy. It is built on the sensitivity defined as follows.

Definition 2. (sensitivity). For any query $f(D)$: $D \to R^d$, l_1-norm sensitivity is the maximum l_1-norm of $f(D) - f(D')$, where D and D' are any two instances in neighboring databases as the following equation holds how to capture the sensitivity of two neighboring databases:

$$S_f = \max_{D,D'} \left\| f(D) - f\left(D'\right) \right\|_1. \tag{2}$$

The Laplace mechanism implements differential private protection by adding noise of Laplace distribution to the query result $\mathrm{Ans}(D) = f(D) + \mathrm{Lap}(\beta)$, where $\beta = S_f/\varepsilon$. As shown above, the concept of differential privacy is generally applied in the joint publishing of compound data. The standard concept makes it unsuitable for applications that involve only one person. In this paper, we propose a more rigorous privacy guarantee with k-anonymity and differential privacy methods.

3.3. Coordinate System. We divide a map into grids where each grid is a state in the Markov model. The users' real locations can be denoted by the state grids in a Markov model. Denote R is the area that includes all the state grids. These areas R can be divided into many spaces $R = \{r_1, r_2, r_3, \cdots, r_i\}$, where each r_i means a unit grid in region R. We set up a spatial coordinate system by which the user's accurate longitude and latitude can be represented as X-axis and Y-axis coordinates. Vector coordinates represent each grid unit, which more clearly shows the user's current position and the corresponding state grid in the Markov model. In Figure 3, all grids have the same size, but in the real world, the sizes of each region are not necessarily equal.

The following example illustrates how to use these grids to denote the user's current location. If the user is located in the area r_6, we denote into this state coordinate system, where $l_u = r_6 = [1, 3]$ with $x = 1$ and $y = 3$. As time goes by, the trajectory of a user's movement is represented by a series of state l_{u_i} in the map coordinate system.

3.4. Location Transition Model and HMM. This paper is based on the study of moving users' trajectories, so we propose to use the random process Markov chain [34–36] to simulate the movement of the user from one point to another under temporal correlations. Other constraints, such as the road networks, can also be captured by the Markov model. The kernel of the Markov model is the state transition matrix. The current state only depends on the transition matrix and its previous moment state. As mentioned before, the user's real locations are unobservable and only known by him. Hence, for the adversaries, the user's movement process is a hidden Markov model.

We use P_t to represent the location of the user at timestamp t. $P_t[r_i] = \Pr\left(a_u^t = r_i\right)$ represents the probability that user u appears in the area r_i at timestamp t. Therefore, we construct the Markov process as follows:

$$\Pr\left(a_u^t = r_i\right) = \Pr\left(a_u^{t_i} = r_i \mid a_u^{t_1} = r_1, a_u^{t_2} = r_2, \cdots, a_u^{t_{i-1}} = r_{i-1}\right). \tag{3}$$

In the first-order Markov model, there is the hypothesis that the transition probability of state and the output probability of observation are only dependent on the current state. So the Markov process can be simplified to

$$\Pr\left(a_u^t = r_i\right) = \Pr\left(a_u^{t_i} = r_i \mid a_u^{t_{i-1}} = r_{i-1}\right). \tag{4}$$

The transition probability,

$$\Pr_u^{r_{ij}} = \Pr\left(a_u^{t_{i+1}} = r_j \mid a_u^t = r_i\right), \tag{5}$$

is the one-step transition probability from timestamp t to $t + 1$. The transition probability satisfies the following properties:

$$\begin{gathered} \Pr_u^{r_{ij}} \geq 0, \\ \sum_{i=0}^{\infty} \Pr_u^{r_{ij}} = 1, \\ j = 1, 2, 3, \cdots. \end{gathered} \tag{6}$$

The sum of the transition probabilities of all possible locations of the user from timestamp t to $t + 1$ is 1. We implement the Markov model on the trajectory of moving users and get $P_t = P_{t-1} * \mathrm{TM}$ to denote the probability

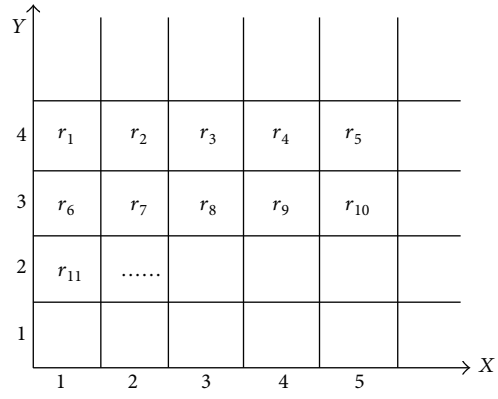


Figure 3: State coordinate system.

transfer at each timestamp. The transition matrix TM is given in our system.

4. UG-Indistinguishable System Model

To apply k-anonymity and differential privacy in the area where moving users share locations on consecutive timestamps, we conduct a rigorous privacy analysis and set a special region R_t (contains an actual user and pseudo users). Even if the adversaries capture this special region, they still cannot identify the target user.

4.1. BB Cloaking Region R. The essence of applying differential privacy in location sharing is to "hide" a real location in a database by adding or removing one record to obtain a "neighboring database." The special region R can be regarded as a set of locations. The adversaries cannot infer whether the target user is in this database R or not with any kinds of queries. However, such a dataset is not completely suitable for our problem. So we proposed a new notion, the black B cloaking region. The black B cloaking region is also noted as a special region to hide users' accurate location at every timestamp. We need to compute the amount of information as follows to obtain the cloaking region at timestamp t:

$$H\left(a_u^t\right) = \sum \Pr\left(a_u^t = r_i\right) \cdot \log\Pr\left(a_u^t = r_i\right), \tag{7}$$

which is also known as the amount of prior information that user u is in region r_i at timestamp t. We intend to use $H(a_u^t = r_i \mid K)$ to represent the posterior information that adversaries infer the user's location information through existing background knowledge K.

$$H\left(a_u^t = r_i \mid K\right) = \sum \Pr\left(a_u^t = r_i \mid K\right) \cdot \log\Pr\left(a_u^t = r_i \mid K\right). \tag{8}$$

In summary, the amount of the user's location information disclosed to the adversary is as follows:

$$H' = H\left(a_u^t\right) - H\left(a_u^t = r_i \mid K\right). \tag{9}$$

Thus, we can obtain a definition of generating a special location set.

Definition 3. (θ-location set). We can set the probability that the adversary can infer the user's current location $\Pr(a_u^t = r_i \mid K)$ as the posterior probability, and the probability of the user at the current location is $\Pr(a_u^t = r_i)$; then, the privacy requirement is as follows:

$$\Pr\left(a_u^t = r_i \mid K\right) - \Pr\left(a_u^t = r_i\right) \le \theta. \tag{10}$$

θ is the privacy threshold of the user's current location at timestamp t and $0 < \theta < 1$. In this article, we set parameter θ as no greater than 0.3. We assume that the parameter θ is given in our framework. When the user's location information exposed to the adversary is greater than the privacy threshold, the cloaking region needs to be generated for protecting the real location at timestamp t.

We define a special region based on k-anonymity and differential privacy, which intuits that the released area will not help an adversary to distinguish any instances in the region. According to Definition 1, we make a transformation that adjusts to a special region R in our article. The new definition is shown as follows.

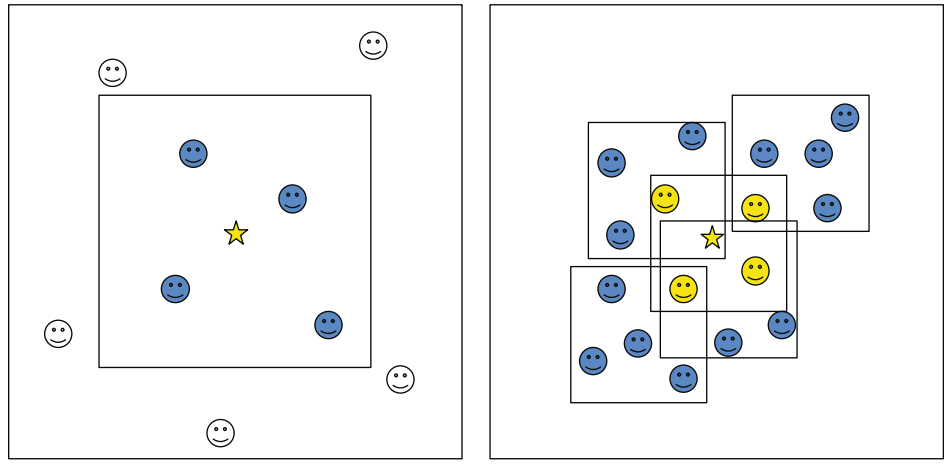
Definition 4. (BB cloaking region differential privacy$*$). At any timestamp$t$, the cloaking region generation by mechanism$M$is represented asR_t, the query function represented as$q()$, and the query result of$q()$on the cloaking region satisfiesε-differential privacy, and the following holds:

$$\Pr\left(q\left(a_{u_1}^t\right) \in R_t\right) \le e^{\varepsilon} \cdot \Pr\left(q\left(a_{u_2}^t\right) \in R_t\right). \tag{11}$$

The definitions guarantee that the accurate location is always protected in a location set Δ at each timestamp. The released region R_t is differentially private at timestamp t for continuous location sharing under temporal correlation. We use the following context to explain how the special region work. In the beginning, a user moves to a new location where he may send a query (e.g., find the nearby restaurant) [37, 38]. At each timestamp, we denote the user's individual information as $A_{u_i}^t = \{\text{time, location, sex, age, query}\}$. The user can be treated as a target user. Then, we assume a mechanism in our system that can obtain a set of k nearest neighbor users with the same query. This set allows the existence of the dummy users, and our system can release the dummy users. Our anonymous generation process is more complicated, and the best effect is verified by experiments when $k = 5$. We regard these four nearest neighbor users as the new target users, respectively. Hence, we have a set of users, as shown in Figure 4.

4.2. Razor Mechanism. After obtaining a set of nearest neighbor users, we propose a new method, Razor Mechanism, to filter the similar terms. We use this mechanism to eliminate users whom we do not want to appear in the nearest neighbor users set.

Even though the adversaries have side knowledge as much as they can have, they still cannot know which nearest neighbor users are generated by the first anonymity. The Razor Mechanism uses the principle of similarity measurement



(a) The first anonymity (b) The second anonymity

Figure 4: Obtain the set of users.

[36] to filter out the pseudo users generated in the first anonymity. In services based on location information, we usually set the distance between locations as a measure of similarity. The similarity measurement of users in the anonymous area is shown in the following formula:

$$\mathrm{Sim}\left(a_{u_i}^t, a_{u_j}^t\right) = \frac{\mathrm{dist}\left(u_i, u_j\right)}{\mathrm{maxdist}\left(R_t\right)}. \tag{12}$$

As shown in Figure 4(a), the data generated in the anonymous data preprocessing stage are removed as noisy data by the Razor Mechanism. Figure 4(b) shows all the remaining data in the special cloaking region, excluding the location coordinate data of yellow dots.

4.3. Drift and Puppet. We use the Razor Mechanism to filter out noisy points that we do not want to appear because the special region contains almost all similar users and possible location information. The target user's information is also eliminated with a small probability (technically, $\mathrm{dist}(u_i, u_j) = \mathrm{maxdist}(R_t)$). This phenomenon is referred to as "drift" and can be solved with the puppet approach in the special region. We use $l_t(\mathrm{lon}, \mathrm{lat})$ and $l_t{}'(\mathrm{lon}', \mathrm{lat}')$ to denote the user's accurate location and puppet, respectively. The definition of the puppet mechanism is shown as follows.

Definition 5. (puppet). A puppet $l_t{}'$ is a cell in the special set which has the closest distance to the target user's location l_t:

$$l_t'\left(\mathrm{lon}', \mathrm{lat}'\right) = \arg\max_{\widehat{l} \in \Delta} \mathrm{dist}\left(\widehat{l}, l\right). \tag{13}$$

In this equation, Δ represents the special location set, and the function $d(\widehat{l}, l)$ denotes the distance between two users in the special region. Note that the puppet approach does not leak any information about the target user. If the target user is in the special set, we protect the target user in the region; otherwise, the puppet is then protected in the special set. Using a puppet does not disclose whether the user's location l_t is in R_t or not. We have mentioned before that our location release mechanism is treated as a black box region. It is still a black box after replacing the accurate location with a puppet.

5. Location and Trajectory Release Algorithm

5.1. Framework. The framework of the special location region release algorithm is shown in Algorithm 1. We generate a special location set at every timestamp to protect a single user's accurate location continuously. The procedure of the generation of the special location set at timestamp t is explained in the context above. First, from lines 1 to 6 in Algorithm 1, the model makes a prediction based on the hidden Markov model. At each timestamp t, we compute the probability p_{t-1}. If the current location at timestamp t satisfied the privacy threshold ($p_{t-1} \le \theta$), the procedure moves to the next timestamp. Otherwise, the special cloaking region is generated at the current location. The process of the special location set is shown in lines 8 to 19. If the target user is filtered out of the special location set, we use a puppet in Δ_t at timestamp t as if it is the "target" user in the release mechanism. Our proposed algorithm uses l_1-norm to capture the sensitivity of the special location set. After all these steps, we can obtain a special location set. According to this special location set, we can generate a special region for a single user at timestamp t. In this region, no matter how much side knowledge the adversaries may have, the adversaries can no longer distinguish the target user from the users.

5.2. Linking Differential Privacy to Trajectory. A user's location trajectory is a moving path or trace reported by a moving object in the geographical space. The user's trajectory T_{a_u} is represented by a set of n time-order points, $T_{a_u} : p_1 \longrightarrow p_2 \longrightarrow \cdots \longrightarrow p_n$, where each point p_i consists of a geospatial coordinate set (X_i, Y_i) and timestamp t (i.e., $p_i = (X_i, Y_i, t)$, where $1 \le i \le n$). Such temporal and spatial attributes of a location trajectory can be considered powerful quasi-

Framework.
Input: a_u^t, TM, p_{t-1}, l, l'
Output: Cloaking region R_t
$p_t \longleftarrow p_t$TM;
Pr $(a_u^t = r_i)$ − Pr $(a_u^t = r_i \mid K) = p_{t-1}$;
if $p_{t-1} \geq \theta$ **then**
    Construct a special set of this location;
else
    Go to next timestamps;
end if
Construct a special location set:
Run k-anonymity$\longrightarrow$set Δ^*;
$r_i = (x_i ; y_i)$;
for (int $k = 0, k \leq 5, k$++) **do**
    r_i = Random $(x_i^* ; y_i^*)$
    if dist$(r_i ; r_i^*) \in$ (dist$_{\min}$; dist$_{\max}$) **then**
        add r_i^* to Δ^*
        algorithm goes on
    else
    go to line 11
    end if
end for
    $\Delta^* \longrightarrow$ Razor Mechanism;
    Sim$(a_{u_i}^t ; a_{u_j}^t)$ = dist$(u_i ; u_j)$/max (R_t)
while Check l' **do**
    if $l \in R_t$ **then**
        algorithm goes on;
    else $\{l \notin R_t\}$
        $l' \longleftarrow$ surrogate;
    end if
end while
    Obtain sensitivity of the special set Δ_1;
    Δ_1 + Lap$(\Delta f/\varepsilon) = R_t$;
    Releases this region R_t;
end;
return Algorithm;
$\longrightarrow$ Go to the next timestamps

Algorithm 1

identifiers that can be linked to various other kinds of physical data objects [39, 40]. From the adversaries' point of view, these trajectories may disclose users' individual information such as users' work, home, and points of interest (POI). Although such trajectories can be made anonymous by replacing the identifier of users with random identifiers, the users may still suffer from privacy threats.

In this paper, our approach uses the Markov model to denote users' movement from one special region to another. We use the equation $p_t = p_{t-1} * \mathrm{TM}$ to denote a single user moving from one region (at timestamp $t-1$) to another region (at timestamp t), and TM denotes the transfer mechanism of users' movement. The transfer mechanism uses Laplace noise to make users' trajectories indistinguishable. As shown in Figure 5, we add Laplace noise to the Markov transfer mechanism to make users' transition probability basically the same. For example, when a user moves from region r_1 to another, according to his habits, the transfer

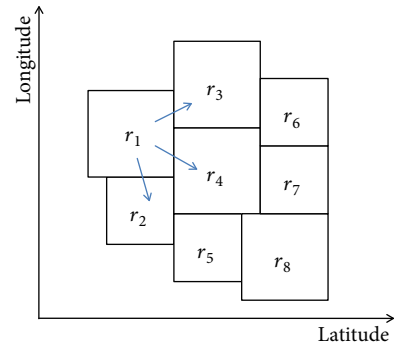


Figure 5: Possible transfer state.

probability at each region is different (e.g., the probability from r_1 to r_2 is 5/10, r_1 to r_3 is 2/10, and r_1 to r_4 is 3/10). According to our method, after adding Laplace noise to the transfer mechanism, the transition probability from r_1 to r_2 is 4/10, r_1 to r_3 is 3/10, and r_1 to r_4 is 3/10. In the following section, we will show the performance by the experiment results.

6. Experiment and Evaluation

In this section, we present the evaluation of our method. All algorithms are implemented in Python on macOS with the real-world datasets GeoLife and Gowalla [41–43]. The GeoLife dataset is collected in (Microsoft Research Asia) GeoLife project by 182 users from April 2007 to August 2012. A time-stamped sequence of points represents the GPS trajectories in this dataset. Each point contains information on latitude, longitude, and altitude. This dataset has 17,621 trajectories with a total distance of 1,292,951 kilometers and a total duration of 50,176 hours. The trajectories are updated at a frequency of every 1 ~ 60 seconds. The Gowalla dataset is collected by Stanford University and is a location-based social networking site where users can share their location information by signing in. The dataset collects a total of 6,442,890 check-in locations and 19,651 check-in information. The check-in data is used to train the Markov model. We implement the proposed model by the following steps.

Step 1. Input the training dataset (Gowalla) to train the Markov model and output the prediction results.

Step 2. If the prediction results we obtain from Step 1 did not satisfy the privacy threshold θ, then we need to generate the special location set by our mechanism at the current timestamp. Otherwise, move to the next timestamp and continue Step 1.

Step 3. Check whether the real location is in the special region or not. In the process of generating a special region, we have a small probability of filtering out the true location. So we use the nearest and the most similar location as a puppet in Δr instead of it.

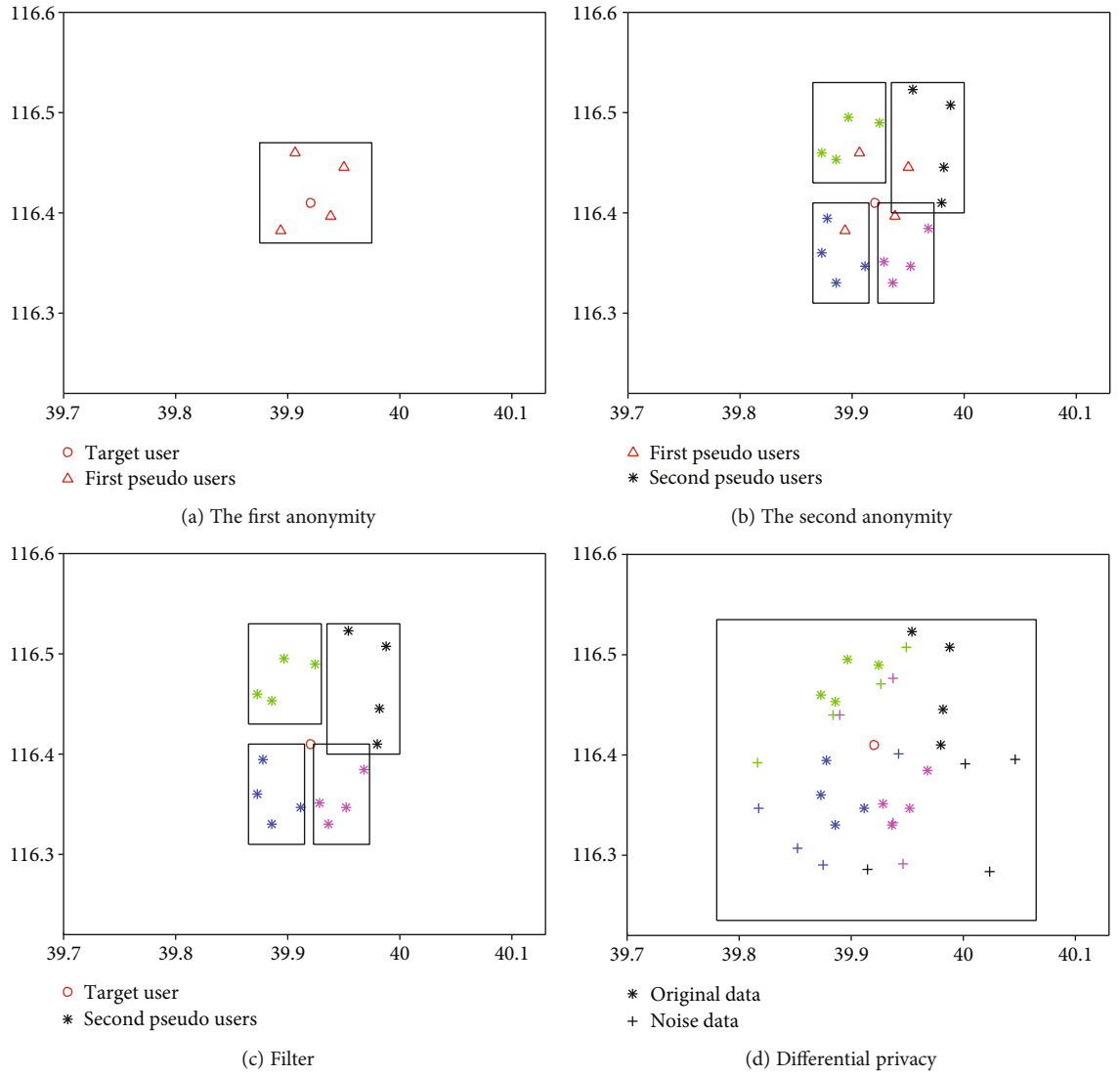


Figure 6: Differential privacy and k-anonymity.

Step 4. Use the special region R_t in each timestamp to divide the real-world map into several neighboring grids. In each grid, the adversaries cannot distinguish between the target user and the pseudo user.

Step 5. Finally, we add noises to the Markov model. In each timestamp, we add noise to the transfer mechanism to make the trajectories indistinguishable.

The performance of the release mechanism as a user moves over time is explained as follows. We treat our release mechanism at each timestamp with $\varepsilon = 1$. Each method is run over 20 times and shows outstanding performance. Figure 6(a) shows how to hide a user's accurate location by the first anonymity. The X-axes and Y-axes represent the longitude and latitude, respectively. The symbol "$\circ$" denotes the user's true location. "Δ" are the pseudo locations generated in the first anonymity. We have the SSE (sum of the squared errors) as the core indicator for the selection of k

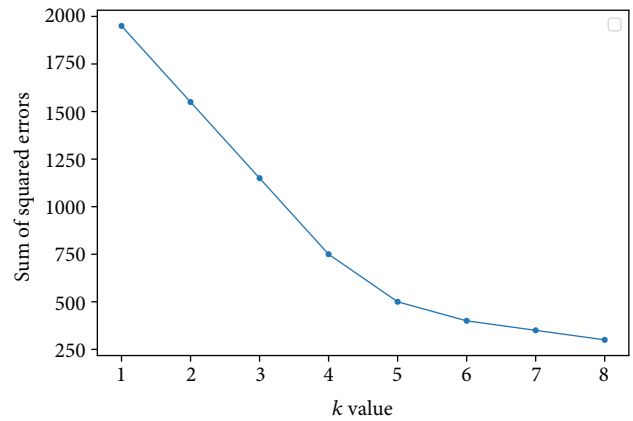
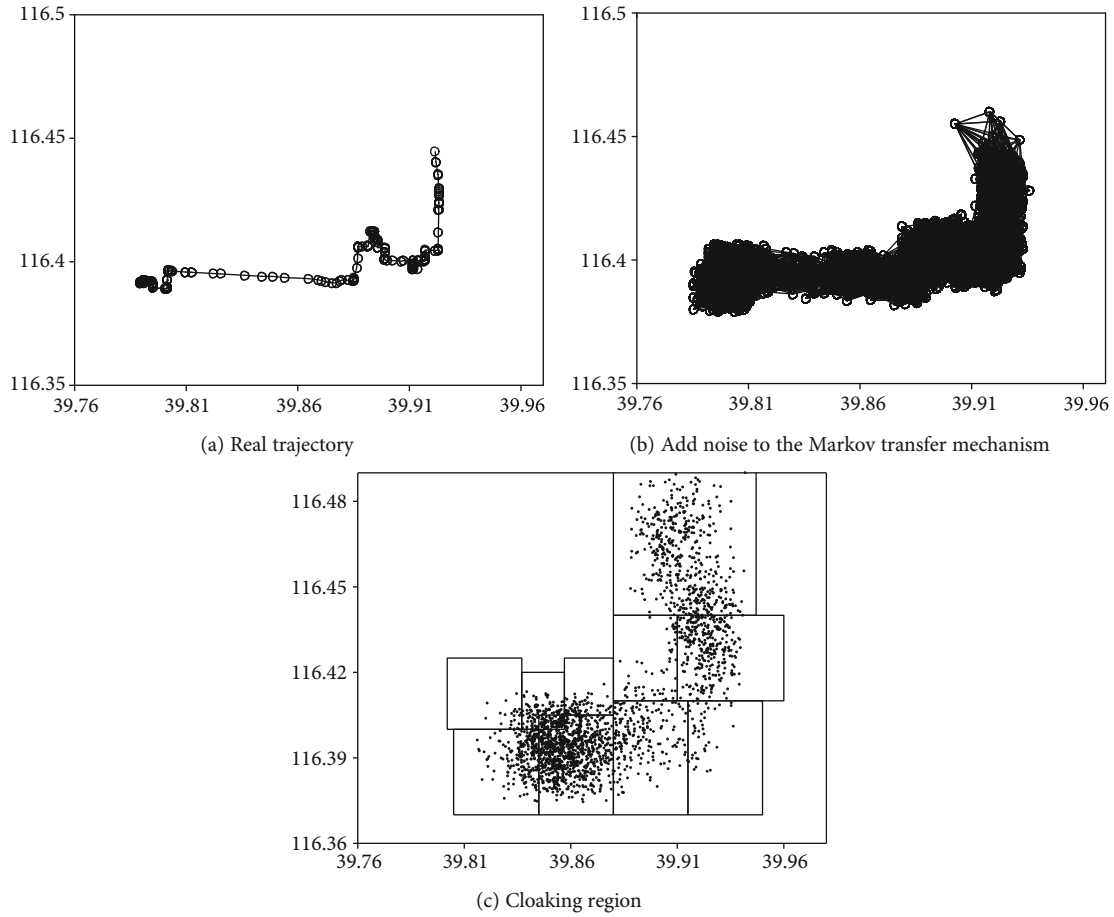


Figure 7: k size vs. sum of the squared errors.



(a) Real trajectory

(b) Add noise to the Markov transfer mechanism

(c) Cloaking region

Figure 8: Compared with the real trajectory.

value. As the k value increases, the sample division gets more refined, and the degree of aggregation of each cloaking region gradually increases. Then, the SSE naturally gradually becomes smaller. In our method, we set the parameter $k = 5$. We have experimented for many times that the parameter k size is better than others, as shown in Figure 7.

We choose four users who are the most similar to the real user and sent the same query at timestamp t. In Figure 6(b), we can see the second k-anonymity after Figure 6(a). In the second anonymity, we consider the first four users generated by the first anonymity as the "real" user, respectively. Then, the model generates more anonymous users by these four "real" users. In Figure 6(b), the symbol "$*$" denotes the pseudo users generated in the second anonymity. Through these two operations, we obtained a much bigger anonymous area with many similar anonymous users. Next, through the "Razor Mechanism" in Section 4, the model filters out several pseudo users generated from the first k-anonymity with "Razor," as shown in Figure 6(c). In this part, we make full use of the Jaccard Razor. While using the "Razor Mechanism," the actual user may be filtered out with a very small probability, which is known as the "drift" phenomenon.

When a "drift" happens by a minuscule probability, we use a surrogate user in ΔR_t to impersonate the target user.

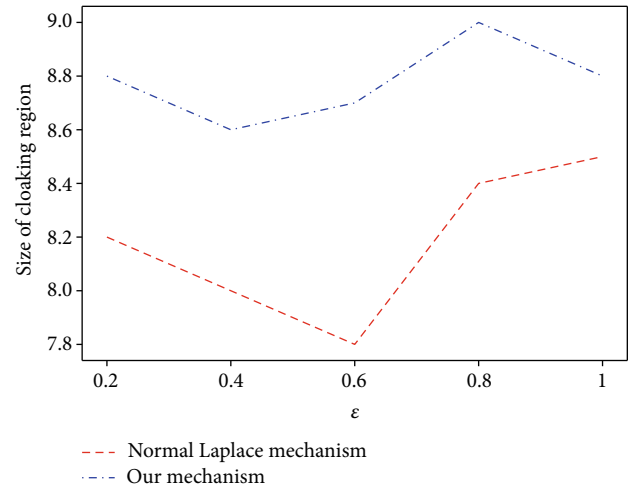
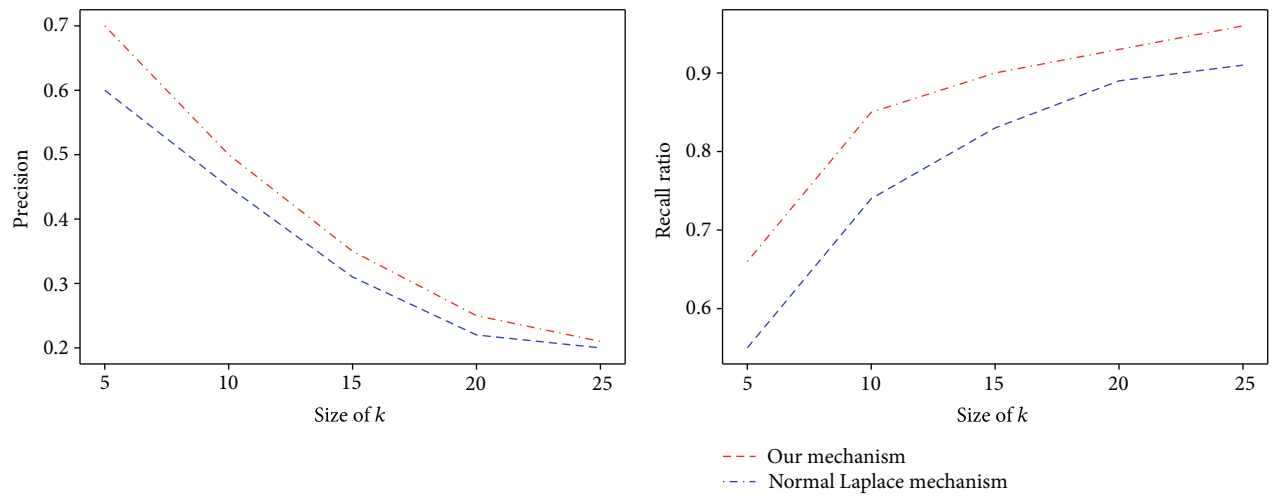


Figure 9: Size of the cloaking region vs. ε.

In the next step, the model adds Laplace noise to this special region ΔR_t at timestamp t, which can provide a rigorous privacy guarantee. We then obtain a new cloaking region that contains the pseudo location and the true location. As shown in Figure 6(d), the area within the square is one of the grids in the real-world maps. The symbol "$*$"



(a) Precision with the size of k

(b) Recall with the size of k

Figure 10: KNN results.

denotes the pseudo users generated in the second anonymity. Finally, the model generates a noisy cloaking region. The added Laplace noise makes the new special region very stable. Noisy users are always around the real user and fake users. In this special region, the adversaries cannot distinguish the target user and the pseudo users. In Figure 8, the true trajectory is compared with the trajectories with noise added to the Markov transfer mechanism. Figure 8(a) is a randomly selected accurate trajectory of a single user in a period of time. The user's movement is shown in Figure 8(a). The most important step in the algorithm is the addition of Laplace noise to the Markov transfer mechanism. This process makes the transfer probability stable. The noisy trajectories after adding noise with the Laplace mechanism are presented in Figure 8(b). As shown in Figures 8(a) and 8(b), the released trajectory is still close to the accurate trajectory. The special regions at every timestamp t are used to divide the real-world map into grids, as shown in Figure 8(c). In this region, the adversaries' side knowledge no longer affects privacy protections. The adversaries cannot either distinguish the accurate trajectory in the released trajectories or recognize the target user in these special regions ΔR_t at each timestamp. Our mechanism is the better one comparing to the normal Laplace mechanism, as shown in Figure 9.

To show the practicality of the release location area, we measure the query accuracy and recall rate of k nearest neighbors for every 500 timestamps in 150 trajectories, as shown in Figure 10. In Figure 10(a), it shows that the precision declines when k rises because when k grows, the nearest neighbors have to be found in larger areas. And a larger location set returned. On the other hand, Figure 10(b) indicates that the recall ratio increases with greater k. Figure 8 shows the comparisons of experiment results by the proposed method's position release mechanism in this paper and those by the Laplace mechanism. The results indicate that the usability of our method is better than that of the Laplace mechanism.

7. Conclusion and Future Work

In this paper, we proposed a L&A-indistinguishable system under temporal correlation. The system uses the Markov model to denote users' movement on the road network and then generates a special user set by k-anonymity and differential privacy approaches. The proposed system can provide perfect privacy protection for a single moving user. The method is based on the hidden Markov model and learns from historical trajectories to obtain prediction results for the future timestamp.

As a direction for future work, we are interested in instantiating the system with different and more advanced mobility models and researching the impact on the system's performance change. We look forward to making the mobile user's personal information protected with a more rigorous privacy guarantee with a smaller loss in data utility. We aim to conduct more profound research to enhance the availability of the region release mechanism based on the existing research studies. We plan to develop a model to recommend points of interest, based on the user's movement position information, and to recommend the community to which the user may move.

Data Availability

The coordinate data used to support the findings of this study have been deposited in the GeoLife dataset repository [44]. The check-in data used to support the findings of this study have been deposited in the Gowalla dataset repository [45].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Our research fund is supported by the National Natural Science Foundation of China (Grant Nos. 61472096, 61272186,

61472095, and 61502410), 2019 Industrial Internet Innovation and Development Engineering, Industrial Internet Security Audit Technology and Product, On-site Emergency Detection Tools in the Field of Industrial Internet Security (KY10600200008, KY10600200021), International Governance Research Center of industrial Internet (3072020CFP0601), and Fundamental Research Funds for the Central Universities (3072020CF0604).

References

[1] Z. Cai and Q. Chen, "Latency-and-coverage aware data aggregation scheduling for multihop battery-free wireless networks," *IEEE Transactions on Wireless Communications*, p. 1, 2020.

[2] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[3] J. Freudiger, R. Shokri, and J.-P. Hubaux, "Evaluating the privacy risk of location-based services," in *Proceedings of the 15th International Conference on Financial Cryptography and Data Security, FC'11*, pp. 31–46, Berlin, Heidelberg, 2012.

[4] J. Freudiger, S. Rane, A. E. Brito, and E. Uzun, "Privacy preserving data quality assessment for high-fidelity data sharing," in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security, WISCS '14*, pp. 21–29, 2014.

[5] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, MobiSys '03*, pp. 31–42, New York, NY, USA, 2003.

[6] Z. Xiong, Z. Cai, Q. Han, A. Alrawais, and W. Li, "Adgan: protect your location privacy in camera data of auto-driving vehicles," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[7] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy," *Proceedings of the VLDB Endowment*, vol. 3, 2010no. 1-2, pp. 619–629, 2010.

[8] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper: query processing for location services without compromising privacy," *ACM Transactions on Database Systems*, vol. 34, pp. 1–48, 2009.

[9] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, pp. 1010–1027, 2001.

[10] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[11] C. Dwork, "Differential privacy in new settings," *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '10*, 2010, pp. 174–183, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2010.

[12] Y. Zak and A. Even, "Development and evaluation of a continuous-time Markov chain model for detecting and handling data currency declines," *Decision Support Systems*, vol. 103, pp. 82–93, 2017.

[13] G. Theodorakopoulos, R. Shokri, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Prolonging the hide-and-seek game: optimal trajectory privacy for location-based services," *Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES '14*, 2014, pp. 73–82, ACM, New York, NY, USA, 2014.

[14] J. Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[15] J. Wang, Z. Cai, and J. Yu, "Achieving personalized k-anonymity-based content privacy for autonomous vehicles in CPS," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4242–4251, 2020.

[16] P. Shankar, V. Ganapathy, and L. Iftode, "Privately querying locationbased services with SybilQuery," in *Proceedings of the 11th International Conference on Ubiquitous Computing, UbiComp '09*, pp. 31–40, New York, NY, USA, 2009.

[17] D. Nussbaum, M. T. Omran, and J.-R. Sack, "Techniques to protect privacy against inference attacks in location based services," *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS '12*, 2012, pp. 58–67, ACM, New York, NY, USA, 2012.

[18] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with PrivacyGrid," in *Proceedings of the 17th International Conference on World Wide Web, WWW '08*, pp. 237–246, New York, NY, USA, 2008.

[19] J. Cuellar, M. Ochoa, and R. Rios, "Indistinguishable regions in geographic privacy," *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC '12*, 2012, pp. 1463–1469, ACM, New York, NY, USA, 2012.

[20] M. Xue, P. Kalnis, and H. K. Pung, "Location diversity: enhanced privacy protection in location based services," *Proceedings of the 4th International Symposium on Location and Context Awareness, LoCA'09*, 2009, pp. 70–87, Springer-Verlag, Berlin, Heidelberg, 2009.

[21] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12*, pp. 617–627, New York, NY, USA, 2012.

[22] M. Herrmann, C. Troncoso, C. Diaz, and B. Preneel, "Optimal sporadic location privacy preserving systems in presence of bandwidth constraints," *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, WPES '13*, 2013, pp. 167–178, ACM, New York, NY, USA, 2013.

[23] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[24] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[25] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: theory meets practice on the map," in *2008 IEEE 24th International Conference on Data Engineering*, pp. 277–286, Cancun, Mexico, 2008.

[26] R. Chen, B. C. Fung, B. C. Desai, and N. M. Sossou, "Differentially private transit data publication: a case study on the Montreal transportation system," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*, pp. 213–221, New York, NY, USA, 2012.

[27] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[28] H. Li, L. Xiong, L. Zhang, and X. Jiang, "DPSynthesizer," *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, vol. 7, 2014no. 13, pp. 1677–1680, 2014.

[29] N. Li, W. Yang, and W. Qardaji, "Differentially private grids for geospatial data," in *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013), ICDE '13*, pp. 757–768, Washington, DC, USA, 2013.

[30] S.-S. Ho and S. Ruan, "Differential privacy for location pattern mining," *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS, SPRINGL '11*, 2011pp. 17–24, New York, NY, USA, 2011.

[31] R. Dewri, "Local differential perturbations: location privacy under approximate knowledge attackers," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2360–2372, 2013.

[32] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[33] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Third Conference on Theory of Cryptography, TCC'06*, pp. 265–284, Berlin, Heidelberg, 2006.

[34] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," *Proceedings of the 2011 IEEE Symposium on Security and Privacy, SP '11*, 2011, pp. 247–262, IEEE Computer Society, Washington, DC, USA, 2011.

[35] M. Götz, S. Nath, and J. Gehrke, "MaskIt: privately releasing user context streams for personalized mobile applications," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12*, pp. 289–300, New York, NY, USA, 2012.

[36] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," *Artificial Intelligence*, vol. 171, no. 5-6, pp. 311–331, 2007.

[37] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, Texas, USA, 2019.

[38] Z. Cai and T. Shi, "Distributed query processing in the edge assisted IoT data monitoring system," *IEEE Internet of Things Journal*, p. 1, 2020.

[39] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: a survey of recent developments," *ACM Computing Surveys (Csur)*, vol. 42, pp. 1–53, 2010.

[40] M. E. Nergiz, M. Atzori, and Y. Saygin, "Towards trajectory anonymization: a generalization-based approach," *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, SPRINGL '08*, 2008, pp. 52–61, ACM, New York, NY, USA, 2008.

[41] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proceedings of the 10th International Conference on Ubiquitous Computing, UbiComp '08*, pp. 312–321, New York, NY, USA, 2008.

[42] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proceedings of the 18th International Conference on World Wide Web, WWW '09*, pp. 791–800, New York, NY, USA, 2009.

[43] Y. Zheng, X. Xie, and W.-Y. Ma, "GeoLife: a collaborative social networking service among user, location and trajectory," *Data Engineering*, vol. 33, p. 32, 2010.

[44] https://www.microsoft.com/en-us/download/details.aspx?id=52367.

[45] http://snap.stanford.edu/data/loc-Gowalla.html.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6630265, 15 pages
https://doi.org/10.1155/2021/6630265

Research Article

TPR-DTVN: A Routing Algorithm in Delay Tolerant Vessel Network Based on Long-Term Trajectory Prediction

Chao Liu,[1] **Yingbin Li**,[1] **Ruobing Jiang**,[1] **Yong Du**,[1] **Qian Lu**,[2] **and Zhongwen Guo**[1]

[1]*Department of Computer Science and Technology, Ocean University of China, China*
[2]*Department of Computer Science and Technology, Qingdao University, China*

Correspondence should be addressed to Ruobing Jiang; jrb@ouc.edu.cn and Yong Du; csyongdu@ouc.edu.cn

Received 19 October 2020; Revised 4 December 2020; Accepted 15 January 2021; Published 29 January 2021

Academic Editor: Xiao Zhang

An efficient and low-cost communication system has great significance in maritime communication, but it faces enormous challenges because of high communication costs, incomplete communication infrastructure, and inefficient routing algorithms. Delay Tolerant Vessel Networks (DTVNs), which can create low-cost communication opportunities among vessels, have recently attracted considerable attention in the academic community. Most existing maritime ad hoc routing algorithms focus on predicting vessels' future contacts by mining coarse-grained social relations or spatial distribution, which has led to poor performance. In this paper, we analyze 3-year trajectory data of 5123 fishery vessels in the China East Sea. Using entropy theory, we observe that the trajectory of the vessel has strongly spatial-temporal distribution regularity, especially when previous states were given. To predict accurate future trajectories, we develop a long-term accurate trajectory prediction model by improving the Bidirectional Long-Short Term Memory (Bi-LSTM) model. Based on predicted trajectories and the confident degree of each prediction step, we propose a series of routing algorithms called TPR-DTVN to achieve efficient communication performance. Finally, we carry out simulation experiments with extensive real data. Compared with existing algorithms, the simulation results show that TPR-DTVN can achieve a higher delivery ratio with lower cost and transmission delay.

1. Introduction

As the critical technology for smart ocean, a low-cost and efficient maritime communication system plays a key role in daily communication [1, 2]. Most maritime activities, such as environmental monitoring, fishery operations, and knowledge exchange, can be successfully implemented under the above communication system [3, 4]. Existing maritime communication systems, including satellite-based, sea-based, and shore-based, have severe shortages in terms of high cost volume, limited network coverage, and absence of facilities [5], which cannot meet the requirements of low cost, wider data transmission in ocean regions [6].

With growing marine operations, the mobile vessel is becoming extremely important in the ocean due to its storage and communication capabilities. In the academic community, considerable attention has been paid in the maritime delay tolerant vessel networks (DTVN) [7–11]. The routing algorithm of DTVN should make a correct decision on each relay based on known or expected information to ensure an effective data transmission. Thus, it is pivotal to get vessel mobility trends and accurately predict future contacts according to historical data [12, 13].

Recent works mainly focus on modeling mobile regularity by setting up the default movement model and mining social relationship [7, 8, 12, 14]. Most of the above work achieves coarse-grained, which is not enough to describe precise mobility patterns. The future trajectory is one of the most vital factors to build an efficient delay tolerant network system [13]. Once given the future trajectory, it is feasible to calculate expected contacts and get an optimal routing plan. Thus, the gap between actual and predicted trajectory has

significant effects on data delivery in DTVN, which means that the accurate trajectory prediction model plays a vital role in the routing process.

Vessels' movement is free space moving due to lack of map topology. Many factors such as fish distribution, weather conditions, and ocean current have significant effects on their movement [15–17], which makes traditional prediction models on land not infeasible for vessel trajectory prediction. For modeling vessels' mobile regularity, some researchers design models based on high-order Markov chain, Bayes model, and deep learning methods [15, 16, 18]. By splitting area to nets, some machine learning models can also be applied by transferring regression to classification problem [19]. The methods based on long-term high-order Markov and Bayes models can predict long-term trajectory. However, the model has low cross-regional extendibility based on our experimental study, which means that the model cannot work when facing a new ocean area. Existing deep learning-based methods all focus on short-term (less than 20mins) trajectory prediction or vessel state estimation. Besides, the regression results do not have proper evaluation metrics on each prediction step, making routing algorithms not adopt it.

To address the above problems, we design a trajectory prediction algorithm to model the vessel's mobile regularity and get the long-term accurate predicted trajectory. Besides, the routing algorithm takes the evaluation result of predicted trajectory as a condition. Even the future trajectories are available, the routing algorithm is essentially a global optimization problem, which is also an NP-hard problem. Therefore, we need to properly design the routing algorithm using predicted future trajectory and evaluation results, to improve the routing performance.

In this paper, we propose a set of low-cost, large-scale DTVN algorithms based on long-term predicted trajectories. Firstly, we utilize entropy to verify the predictability of vessels with massive actual traces data. It can be observed that the regularity in trajectory has lower entropy with more previous states and sailing features. Then, we design an improved Bi-LSTM trajectory predict model with great cross-regional extendibility to model the mobile regularity. To address the predicted trajectory evaluation problem, we utilize the prediction model's performance on the test set to design the confident degree of each predicted step calculation method. Finally, a series of low-cost, large-scale DTVN routing algorithms based on the predicted trajectory is designed.

We have made the following intellectual contributions in this paper:

(1) It is the first work, to the best of our knowledge, to exploit the precise mobility pattern of vessel networks and use them for routing algorithms design in the vessel maritime communication system

(2) We design an accurate long-term trajectory prediction model based on improved Bi-LSTM to model vessels' mobility patterns, which achieves more than 9-hour accurate predictions. The trajectory prediction model shows an excellent prediction performance even the input sequences have never appeared in training sets. It indicates that our model can achieve a cross-regional trajectory prediction

(3) An advanced evaluation method on trajectory regression results is designed to solve the regression model evaluation problem, making the predicted trajectory fully used by our DTVN routing algorithms

(4) We propose a series of routing algorithms based on predicted trajectory to solve the optimal routing problem, which proves to be NP-Hard. Simulation in extensive actual trajectory data set, including 5123 vessels in 3 years, shows that our routing algorithm can achieve a higher delivery ratio than other routing strategies with lower cost and delay

The rest of the paper is organized as follows. The system model and problem formulation are given in Section 2. Section 3 introduces our basic idea and challenges. The structure of our trajectory prediction model is proposed in Section 4. The routing algorithm based on trajectory prediction is presented in Section 5. Section 6 gives the introduction of distributed algorithms. Section 7 shows the performance of our algorithm. Section 8 reviews related works. This paper is concluded in Section 9.

2. System Model and Problem Formulation

In this section, we introduce the system model of DTVN. Then, we describe the optimal target of system, expected delivery probability, and system evaluation metrics including system cost, delivery delay, delivery ratio, and system efficiency. Finally, we prove that the system optimization problem is NP-Hard.

2.1. System Model. All vessels in DTVN are considered to be a set of nodes represented by S. Node trajectory s in S is specified as a sequence of states over a period of time $(t1, tn)$

$$T_s = \{c_s, t_1, c_s, t_2, \cdots, c_s, t_n\}, s \in S. \tag{1}$$

In T_s, each state $c_{s,t}$ has its information, including latitude, longitude, speed, and direction. For nodes s_i and s_j in S, we use $Dis(s_i, s_j)$ to denote the distance between them and utilize r_{si}, r_{sj} to reflect their communication radius. Therefore, s_i and s_j can form a communication link when $\mathrm{Dis}(s_i, s_j)$ is smaller than min (r_{si}, r_{sj}). The whole set S can form a link set at the time t, which can be specified

$$\begin{aligned} L_t &= \left\{ \left(s_i, s_j\right) \middle| s_i, s_j \in S \right\}, \\ \mathrm{Dis}&\left(p_{s_i,t}, p_{s_j,t}\right) < \min\left(r_{s_i,r_{s_j}}\right). \end{aligned} \tag{2}$$

For a packet p, we use $\delta(p)$ and $\zeta(p)$ to denote its source and destination. The p could be copied for transfer to the node when they have links between them. Then, the packet

set P_n of size n is defined as

$$P_n = \{p_i(\delta(p_i), \zeta(p_i)) | \delta(p), \zeta(p) \in S\}, \quad i = 1, 2, \cdots, n. \tag{3}$$

We also set TTL (time-to-live) for the packet's remaining living time, $\mathrm{TTL}_{\max}$ for the maximum living time, H for remaining hops, and $H_{\max}$ for maximum hops. Packet delivery will fail if p_i does not get to its destination under the situation that TTL or H is equal to 0.

We next give the formal definitions of four system evaluation metrics system cost, delivery delay, delivery ratio, and system efficiency.

Definition 1 (system cost). System cost represents the network occupation in the packet delivery process. In our system, $\phi(p)$ is defined as the number of copies generated in the delivery process of p. Therefore, the system cost of P_n can be given by

$$\phi(P_n) = \sum_{i=1}^{n} \phi(p_i), p_i \in P_n. \tag{4}$$

Definition 2 (delivery delay). To measure the delivery time of p from source $\delta(p)$ to destination $\zeta(p)$, $\chi(p)$ is denoted as the delivery delay of p. $t(\delta(p))$ is defined as the timestamp of p at the source node. So $t(\zeta(p))$ is at the destination node. Delivery delay of p can be defined as $\chi(p) = t(\zeta(p)) - t(\delta(p))$. The system delivery delay is the average delay of P_n, which can be given by

$$x(P_n) = \frac{1}{n} \sum_{p_i \in P_n} x(p_i) \tag{5}$$

Definition 3 (delivery ratio). Delivery ratio is defined as the proportion of the packets successfully delivered to total packets number n. The delivery ratio can be given as $(|p \text{ success}|/n)$.

Definition 4 (system efficiency). A high-performance communication system should have a high delivery ratio with low cost and delay. Hence, we define system efficiency as the proportion of the delivery ratio to system cost multiplying delivery delay. The efficiency of P_n is defined as

$$\text{Efficiency}(P_n) = \frac{|p_{\text{success}}|}{n\chi(P_n) \times \phi(P_n)} \tag{6}$$

2.2. Problem Formulation. The target of the system is to deliver all packets from sources to destinations, respectively. Our system's objective is to maximize system expected delivery probability. Several definitions are given as follows.

Definition 5 (packet expected delivery probability). Given a packet p with its source $\delta(p)$ and destination $\zeta(p)$, the packet expected delivery probability ϱ_p is the expected delivery probability of p, which can be calculated by a function $f()$. ϱ_p can be given by

$$\varrho_p = f(\delta(p), \zeta(p)), \delta(p), \zeta(p) \in S. \tag{7}$$

ϱ_p is a fixed value if all the future trajectories T can be obtained. Additionally, it can be changed with nodes' different predicted future trajectories over time.

Definition 6 (system expected delivery probability). Each packet p in P_n can get ϱ_p at the current time, the system expected delivery probability would be defined as $\sum p \in Pn\, \varrho p$.

The goal of our system is to maximize the system expected delivery probability. In conclusion, the whole optimal target of the system can be given by

$$\max \sum_{p \in P_n} \varrho_p. \tag{8}$$

2.3. Complexity Analysis. To address the expected delivery probability maximization problem, the algorithm needs to choose proper communication paths and a reasonable spectrum resource allocation strategy. The whole optimal problem is NP-hard due to its simplification problem. One can reduce the optimal resource allocation problem to the weighted maximum independent set problem (i.e., NP-hard problem).

Note that the links within interference distance cannot allocate the same spectrum resources, which might generate mutual-interference. A portion of links within the network L_t can allocate the same channel resources, constituting the reuse set. We use $\{\omega_n\}$ to represent all possible reuse sets in L_t. Assuming that all communication paths are already known, and all reuse sets are allocated the same amount of subchannel resources.

For reuse set ω_n, we select the links with a minimum data rate in one path and use κ_n to represent the sum of these links. Then, we let the vertex weight equal to the sum of the reuse sets' data rate. Edges between vertexes represent different reuse sets that have partially repeated links (i.e., no edge between two vertexes represents that two reuse sets are totally different). Even knowing $\{\kappa_n\}$ and completed trajectory $T_i \in S$, this reduction can also be established. So, the weighted maximum independent set problem means to find reuse sets, which has a maximum sum data rate without repeated links.

For any instance I of the weighted maximum independent set problem, we can construct the instance $I*$ of the optimal resource allocation problem by using the partitioning $\omega \in \{\omega_n\}$ to vertex set $\nu \in \{\nu_n\}$ in I and using κ as weight for vertex ν_n. So, if different ω has partially repeated links in $I*$, there are edges between them. The reduction can be accomplished in polynomial time, and it can verify that a feasible solution in I is a feasible solution.

3. Basic Idea and Challenges

In this section, we first verify the basic idea and algorithm feasibility. Also, challenges will be listed during the process of feasibility validation.

3.1. Basic Idea. In Section 2, we formulate the target of our algorithm. From problem formulation, we observe that the essential problem is how to predict future contacts among vessels. Obviously, if future trajectories can be reached or accurately predicted, we may measure the future contacts of the vessels in these trajectories. As a consequence, the DTVN routing problem can be converted into a trajectory forecast and predicted trajectory evaluation. The basic idea for our routing algorithm can be divided into four steps. The basic idea of our routing algorithm can be divided into four steps:

Step 1: long-term precise future trajectories should be predicted by applying a proper algorithm to massive historical data of vessels

Step 2: to make trajectories that can be used by the routing algorithm, predicted trajectories should be evaluated by specially designed measurement methods

Step 3: based on predicted trajectories and their evaluation results, our system calculates the future contacts' probabilities

Step 4: the routing algorithm utilizes contacts' probabilities to make a proper decision on each hop. A simple channel arrangement method should be applied during the data transmission process to solve the global optimal routing problem, which has been proved to be NP-Hard

3.2. Feasibility Validation. The effectiveness of the algorithm depends on whether the future trajectory of vessels can be predicted. To validate the predictability of the vessel's trajectory, we use conditional entropy, which has been commonly used to measure the disorder degree of time series data. Conditional entropy describes the disorder degree in the situation that some previous states have been given. The data is more predictable when it has lower conditional entropy than others.

From Section 2, we know that a vessel has $T_s = \{c_{s,t1}, c_{s,t2}, \cdots, c_{s,tn}\}, s \in S$. Firstly, we need to discretize the data in T_s. Meanwhile, we set the location into 500 m × 500 m girds among the ocean and use the center coordinate to indicate each sample's location. Speed and direction are, respectively, discretized by 1 km and 20°. For all the c_j of T_s, we count how many times c_j occurs and denote it as o_{cj}, where $1 \le j \le (\mathrm{LOC} \times \mathrm{SPE} \times \mathrm{DIR})$:

(i) LOC represents the gird number

(ii) SPE represents the discretized speed number

(iii) DIR represents the discretized direction number

In terms of frequency (o_{cj}/n), we calculate the marginal entropy $H(T_s)$ of T_s by equation (9)

$$H(T_s) = \sum_{j=1}^{\mathrm{LOC}\times\mathrm{SPE}\times\mathrm{DIR}} \frac{O_{c_j}}{n} \times \log_2 \frac{1}{O_{c_j}/n}. \tag{9}$$

Extend T_s to asequence of two-tuples $T_s{}^1 = \{(c_1, c_2), (c_2, c_3), \cdots, {}^1(c_n - 1, c_n)\}$, count how many times (c_α, c_β) occurs in T_s and denote it as $o\alpha, \beta$, finally get joint $H(T_s{}^1, T_s)$ entropy as equation (10)

$$H\left(T_s^1, T_s\right) = \sum_{\forall 1\le\alpha,\beta\le(\mathrm{LOC}\times\mathrm{SPE}\times\mathrm{DIR})} \frac{o_{\alpha,\beta}}{n-1} \times \log_2 \frac{1}{O_{\alpha,\beta}/(n-1)}. \tag{10}$$

Then, conditional entropy $H(T_s{}^1 \mid T_s)$ of $T_s{}^1$ can be calculated by equation (11)

$$H\left(T_s{}^1 \mid T_i\right) = H\left(T_s{}^1, T_s\right) - H(T_s). \tag{11}$$

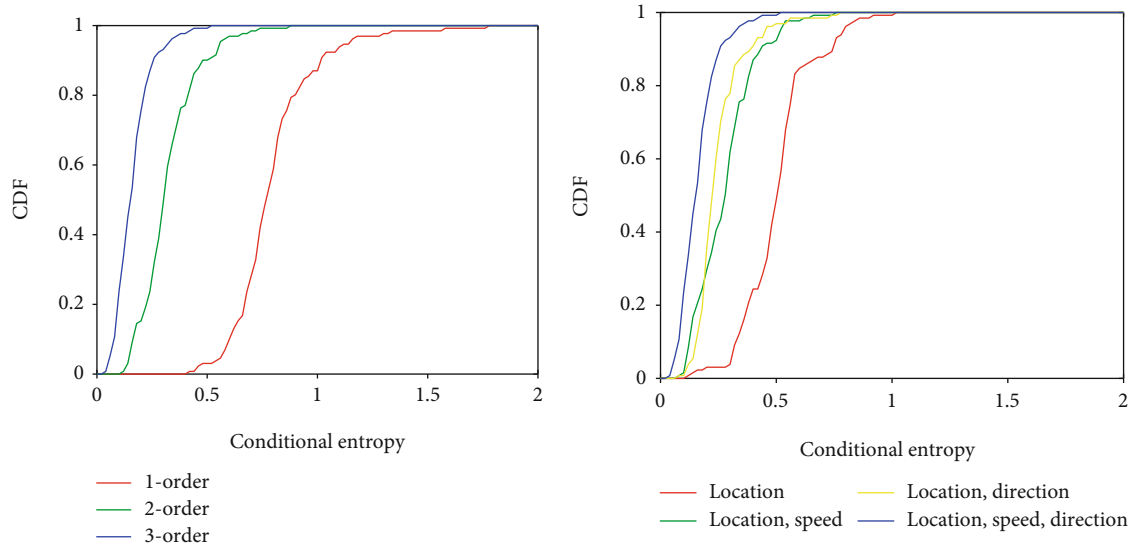
Based on previous equations, we can keep calculating the conditional entropy of T_i^k, and eventually, the conditional entropy of T_i^k is derived as equation (12).

$$H\left(T_s^k \middle| T_s T_s^1 \cdots T_s^{k-1}\right) = H\left(T_s^k \middle| T_s T_s^1 \cdots T_s^{k-1}\right) - H\left(T_s^k \middle| T_s T_s^1 \cdots T_s^{k-2}\right) - \cdots H(T_s). \tag{12}$$

We use entropy theory to analyze our data set. The data set from the Vessel Monitor System in China East Sea, which collects fishery vessel states from May 2015 to May 2018. We select 5123 vessels with good data integrity. System records vessels' sailing features including ship ID, latitude, longitude, speed, and direction on every 3 minutes. The whole data set has a range of 120°E to 130°E and 25°N to 35°N. Before analyzing, we sort all samples in time, remove duplicate and abnormal samples, to avoid any negative effects. Moreover, the local average interpolation method is applied to complete missing samples.

Figure 1(a) shows that the vessel's trajectory uncertainty will decrease when more previous states are given. It illustrates that more historical data will help the mining mobility pattern of vessels. We also analyze the mutual effect of conditional entropy between features. We use different combinations of features, including location, speed, and direction, to measure whether the trajectory is more predictable with different feature combinations. Figure 1(b) illustrates that these additional features are of great benefit to regularity mining. To sum up, it is highly possible that vessel's future trajectory can be predicted by specifically mining historical data.

3.3. Challenges. For time-series data prediction, some traditional models like Bayes or Markov chain have been applied [17, 18, 20]. Based on entropy results, a trajectory prediction model based on the high-order Markov chain has been realized. We use data from May 2015 to May 2016 to train the state transition matrix and the rest for testing. The test results of average continuous prediction time (ACPT) are shown in Table 1. It is obvious that 3-order Markov model with location, speed, and direction could only continuously predict the trajectory for 15.4 minutes on average. The continuous



(a) Different k

(b) $k = 3$ with different features

Figure 1: Conditional entropy CDF with different k and features.

Table 1: Average continuous prediction time (ACPT) of 3-order Markov model.

Features	ACPT (minutes)
Location	46.3
Location, speed	27.7
Location, direction	32.2
Location, speed, direction	15.4

prediction time can improve to 46.3 minutes on average, with 49.7% more distance bias when only applying the location feature. There are 32.2% data in the test set that cannot apply this model, because these models can only work when current states are in the training set. A trajectory cannot be predicted when vessels change their working area, type, or time. Moreover, the Markov model's continuous prediction time and availability will continue decreasing with higher-order, limiting the usage of historical data. Therefore, traditional models like Bayes or Markov chain have shortages on both long-term prediction and cross-regional extendibility. The first challenge is listed as follows:

Challenge 1: on account of traditional models' shortages, a novel trajectory prediction model needs to be designed to solve the historical data usage problems, long-term continuous prediction, and cross-regional extendibility. Section 4 addresses this challenge.

Since vessels' location, speed, and direction are all continuous variables, it is more suitable to use the regression model and sliding window to predict trajectory continuously. However, the regression results do not have a proper metric to evaluate each predicted step. The routing algorithm cannot make relay decisions without assessment criteria. So, challenge 2 and 3 are listed as follows:

Challenge 2: a metric that can approximately represent the probability that the vessel appeared at a predicted location, which should be calculated using a novel evaluation method. The evaluation result needs to be directly adopted by the routing algorithm. This challenge is addressed in Section 5.1.

Challenge 3: a series of distributed routing algorithms with channel arrangements need to be proposed utilizing a novel evaluation method and predicted trajectories. The global optimal routing problem is NP-Hard, according to Section 2.3. This challenge is addressed in Section 5.2, Section 5.3, and Section 6.

4. Trajectory Prediction Model

In this section, we present our trajectory prediction model based on Bi-LSTM in detail. Firstly, the advantages of Bi-LSTM for time-series data are briefly introduced. Then, the detail of our model and its performance are given. Finally, we verify the cross-region extendability of our model.

4.1. Bi-LSTM for Time-Series Prediction. Neural networks have a powerful ability to fit most functions. According to the universal approximation theorem, even a single hidden layer with no linear activation function neural network can approach any continuous finite dimension function with high precision.

For series data, many studies show that Recurrent Neural Network, like LSTM and GRU, has good learning performance to sequence learning problem [21, 22]. To solve the vanishing gradient and exploding gradient problems that might happen in RNN, the LSTM adds a cell, an input gate, a forget gate, and an output gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. Some researchers propose that the Bidirectional LSTM shows a more robust learning ability in sequence data learning [20]. Different from traditional LSTM, one Bi-LSTM unit includes two LSTM units. They can control the forward and backward

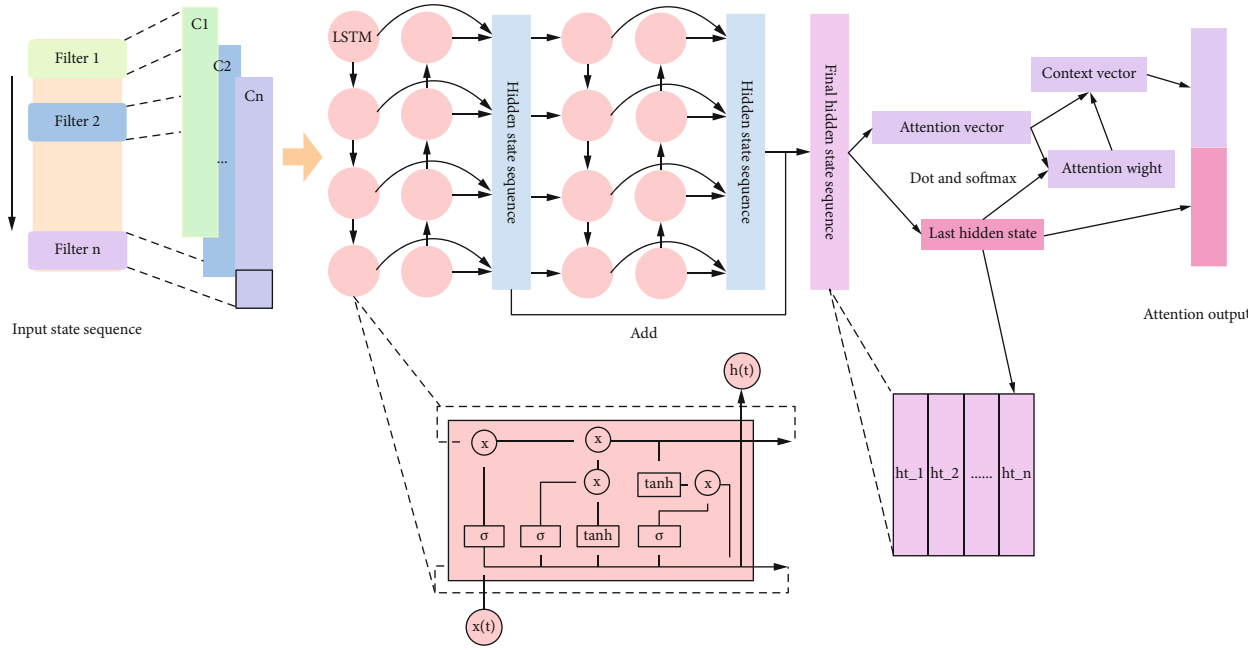


Figure 2: The structure of trajectory predict model and Bi-LSTM.

information flow in their cell state, respectively. In Figure 2, we could notice that the input state in the Bi-LSTM unit can flow in two directions. The unit in the next layer has connected with forward and backward units, which make it have a broader receptive field. Compared with LSTM, the Bidirectional LSTM is more suitable to predict the long-term trajectory in the ocean field.

4.2. Model Design and Performance. Our dataset's raw data is the vessel's state every 3 minutes in 3 years, including time, ship ID, latitude, longitude, speed, and direction. To train the neural network, we first need to get the training sample by transferring the time series to the supervised learning problem. We need to get the input state sequence and output sequence from the original series of states. Assuming the length proportion between the input and output sequence is m, and the length of the predicted sequence is $\mathfrak{I}$. For given state sequence $\{c_1, c_2, \cdots, c_n\}$, the sample$_i$ in sample set is the following:

$$\begin{aligned}\text{Sample}_i = \{x_i : <c_i, \cdots, c_i + m\mathfrak{I} - 1>, y_i : <c_i + m\mathfrak{I}, \cdots, c_i \\ + (m+1)\mathfrak{I} - 1>\},\end{aligned} \tag{13}$$

where $1 \le i \le n - (m+1)\mathfrak{I} + 1$. In this paper, we set $m = 3$ and $\mathfrak{I} = 1$ h in our model, which means that the model utilizes 3-hour historical information to predict the future 1-hour information. In addition to the location, our model also gives the prediction results of speed and direction. Therefore, if TTL is more than 1 hour, we can utilize the sliding window for long-term trajectory predictions. We shuffle the sample set and split the training, validation, and test sets according to the ratio of 7 : 2 : 1. Our model applies normalization to avoid the negative effect of feature range in the learning process.

Figure 2 illustrates our model structure. Our model has four important parts, including convolution, Bi-LSTM, attention, and dense layers. We apply the stack of 1D convolutional layers at the beginning. The convolutional layer can process the grid structure data, including time series as 1D grids or graph as 2D grids. Because of the translation equivariant of convolution kernel, this layer helps our model learn spatiotemporal invariant patterns and reduce its capacity. The following layers also can get a bigger receptive field by the stack of the convolutional layers. As we know, one convolution kernel can only learn one pattern. Different convolution parameters like stride and activation functions have a significant impact on the model's performance. Therefore, we stack 3 layers of convolution with 64, 32, and 16 kernels with *ReLU* and set the stride = 3 in our improved model.

We use Bi-LSTM layers after convolution parts to avoid gradient explosion or vanish, which may lead to unstable state problems during the training process. Benefit from gate units and cell state, the typical LSTM has the memory of historical steps and stable gradient. Compared with typical LSTM, the Bi-LSTM has a more powerful fitting ability in sequence prediction. In each time step, the output of Bi-LSTM considers both forward and backward information. We choose four layers of Bi-LSTM with 64 cells and use *softsign* as an activation function. The *softsign* is better than *tanh* in this task, because it has a more gentle gradient performance than *tanh*. Additionally, we also choose proper weight initialization values based on the activation function. In this part, some residual connections between Bi-LSTM layers are added to avoid gradient vanishment.

The attention mechanism is added after the Bi-LSTM part. The attention mechanism can be used in time series prediction problems and show great performance in many

datasets. The attention, along with time steps, can make the model focus on some important sequences. In this scenario, the *latitude* and *longitude* are more important and have more variability than *speed* and *direction*. The model will treat all features equally if we do not use attention.

As shown in Figure 2, the attention layer gets the attention score by comparing the similarity of the last-hidden state with all history hidden states by the scaled-dot. Considering the *Last hidden state*, it has information learned from the whole input sequence, but the information from the beginning of the input sequence might disappear through long time steps. Therefore, one of the meanings that we design the attention layer is to integrate the information from the whole sequence. After that, *softmax* will be utilized to normalize the score. So, *the context vector* will be the weighted sum of attention score and hidden historical states. Then, we concat *the context vector* and *Last hidden state* as the output of attention layer.

$$\text{score}_i = \frac{\text{hidden}_{\text{last}} \cdot \text{hidden}_i}{\sqrt{\dim_{\text{hidden}}}}, i = 1, \cdots, m\mathfrak{F} - 1. \quad (14)$$

At the end of the model, we add dense layers after the attention layer to obtain the output. In parameter setting, we stack two dense layers with $8\mathbb{J}$ and $4\mathbb{J}$ cells with *ReLU* and *Linear* activation function, respectively. *Huber*(), which has excellent robustness to outlier point, is adopted as the network's loss function. We mainly set the callback, including early stopping and learning rate decay. The model will stop training if *MAE* do not reduce in 30 epochs. We use 0.001 as the learning rate at the first ten epochs and use exponential learning rate decay. Finally, we set a total of 500 epochs to train the model with previous callbacks.

Figure 3(a) indicates the average error on the test set. We can notice that our model can predict future trajectory in 1 hour with an average error of less than 300 meters. To further check the efficiency of our model, we use the sliding window to estimate the 9-hour trajectory. The model could give a longer predicted state sequence using slide window but might reduce the performance because using inaccuracy predicted states as input. And it also requires the model to predict all the features in state, which increases the complexity of trajectory prediction. The result shows that our model can achieve a 9-hour trajectory prediction with an average error of less than 2.73 kilometers.

4.3. Cross-Region Extendability Validation. As mentioned in Challenge 1, traditional models do not have cross-region extendability. To verify our model's extendability, we artificially divide the data set based on each vessel's activity range. We use samples within 60% distance from port to farthest point for model training and the rest for testing. Figure 3(b) shows that the test set's average error is 30% larger than that in Figure 3(a). Compared with long communication distance(average 30 km), this error can be tolerated. The possible reason for this extendability is that our model learns some patterns from other state information, like *speed* and *direction*. Also, the stack of convolutional layers makes our model learn the spatiotemporal invariant patterns. In conclusion, cross-region extendability improves robustness during the applying process.

5. Global Routing Algorithm

In this section, we design the DTVN routing algorithm with global knowledge [**?**]. Like conventional delay-tolerant networks, vessels with larger expected encounter probability are more likely to meet in the future. When two vessels are within their range of contact, the packet should be copied and forwarded to the vessel with higher expected delivery probability in H hops. Therefore, the expected delivery probability calculation is the crucial issue of this algorithm. The process of expected delivery probability calculation is introduced in Sections 5.1 and 5.2. Section 5.3 will give a comprehensive overview of the routing algorithm.

5.1. Confident Degree of Predicted Trajectory

Definition 7 (confident degree of predicted trajectory). Confident degree of predicted trajectory approximately represents the probability that the vessel is on its expected position.

Since the regression results of the improved Bi-LSTM model do not include such metrics, to evaluate the results of predicted trajectories, we propose the confident degree calculation method based on the model's performance on the test set.

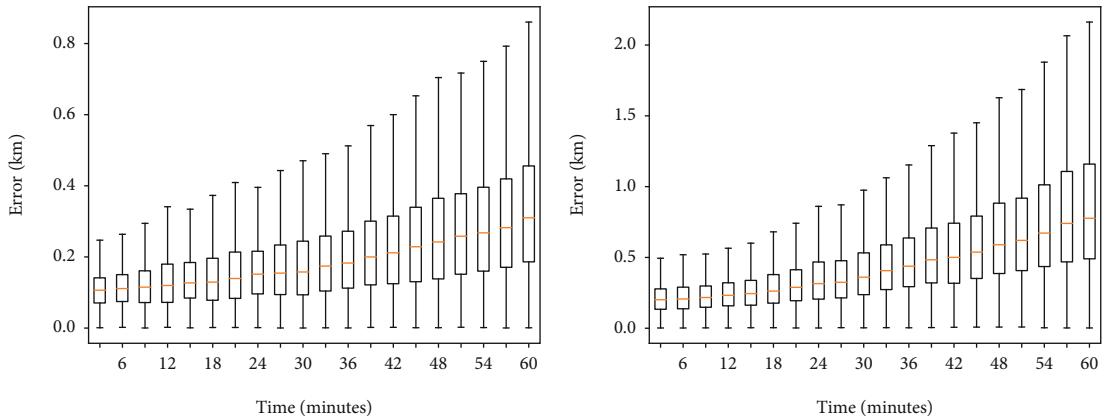
The mean error μ_n and standard deviation σ_n for each prediction step n on test set can be obtained after the model's training process. μ_n and σ_n can show the model's performance on historical data, which helps to determine the expected effects. For all prior assumed distributions, if we have a mean and standard deviation, the normal distribution has the highest entropy, which means that it shall have the highest adaptability. So we choose a normal distribution to fit the error of each predicted step.

The error_n, which is the errors of step n, is denoted as the random variable. We could assume that $\text{error}_n \sim N(\mu_n, {\sigma_n}^2)$. The Probability Density Function (PDF) with μ_n and σ_n can be given by

$$f_n(\text{error}_n) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-(\text{error}_n - \mu_n)^2/2\mu_n^2} dx. \quad (15)$$

Then, $\mathcal{N}(\mu_n', {\sigma'}_n^2)$ can be fitted according to the results of equation (14). Therefore, the probability of $error_n$ less than $(\mu_n' + 3\sigma_n')$ is 99.87%. However, the predicted error can be tolerated in the routing scenario when the communication radius R is large enough. So the confident degree of the predicted trajectory should not only relate with μ_n' and σ_n', but also relate with R. Considering that vessels are moving in 2D space, in this paper, the confident degree of predicted trajectory on step n can be given by

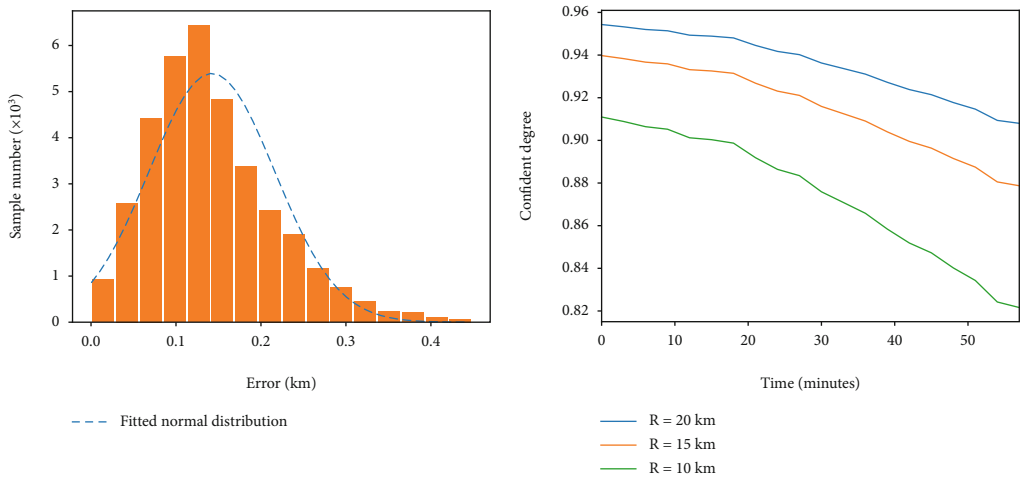
$$E_{v,n} = 99.87\% \frac{\left(R - \left(\mu_n' + 3\sigma_n'\right)\right)^2}{R_2}, \left(\mu_n' + 3\sigma_n'\right) \le R, \quad (16)$$



(a) Test error in random data set

(b) Test error of cross-region extendability validation

Figure 3: Test error in different partitions of data set.



(a) Error distribution on step 11

(b) Confident degree results

Figure 4: Confident degree of vessel "Zhe31828".

where υ is the identifier of vessel υ. When $(\mu_n' + 3\sigma_n') \geq R$, we define $E_{\upsilon,n} = 0$, because the error is too large to guarantee communication. The $E_{\upsilon,n}$ could also be denoted as $E_{\upsilon,t}$ where t is the timestamp of step n. $E_{\upsilon,t}$ can be approximately treated as the probability of vessel appeared on predicted location based on empirical error and communication radius.

Figure 4(a) shows the distribution of the error in the 11th step of the vessel "Zhe31828" and its fitted distribution. We can see that the distribution of the error is fitted to the normal distribution. Confident degree results are shown in Figure 4(b). We can observe that the confident degree changes with different communication radius. The confident degree declines in long-term prediction. The vessel with a smaller communication radius has a lower confident degree due to its tolerance for error.

5.2. Calculation of Packet Delivery Probability within n Hops. For each vessel υ at time t, the confident degree of predicted trajectory $E_{\upsilon,t}$ can be obtained. We use υ and τ to denote two nodes in the ocean. $\varrho^n\upsilon, \tau(t_{\text{start}}, t_{\text{end}})$ stands for expected delivery probability in time period $[t_{\text{start}}, t_{\text{end}}]$ from υ to τ at just n hops, which means $\varrho^n\upsilon, \tau(t_{\text{start}}, t_{\text{end}})$ does not include the situation that packet is delivered in less than n hops.

When υ and τ are expected to meet (within their communication range) at time t, the one-hop delivery probability is $E_{\upsilon,t} \times E_{\tau,t}$. So, the one-hop delivery probability can be given by

$$\varrho^1_{\upsilon,\tau}(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t_\epsilon[t_{\text{start}}, t_{\text{end}}]} (1 - E_{\upsilon,t} \times E\tau, t), \tag{17}$$

Where t is the start time of expected contacts. The accompanying time is not considered in this equation to avoid repeated calculation. The meaning of t is the same in the following derivation procedures in this subsection.

The condition of two-hop delivery can be treated as two one-hop delivery procedures with the time constraint. The

delivery probability of two-hops can be calculated as follows:

$$\varrho^{2}_{v,\tau}(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t_{\text{start}} \le t \le t_{\text{end}}} \left(1 - \varrho^{1}_{v,m}(t_{\text{start}}, t) \times \varrho^{1}_{m,\tau}(t, t_{\text{end}})\right). \tag{18}$$

Then, three-hop delivery probability can be divided into one one-hop delivery procedure and one two-hop delivery.

The three-hop delivery probability can be derived by

$$\varrho^{3}_{v,\tau}(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t_{\text{start}} \le t \le t_{\text{end}}} \left(1 - \varrho^{1}_{v,m}(t_{\text{start}}, t) \times \varrho^{2}_{m,\tau}(t, t_{\text{end}})\right). \tag{19}$$

Therefore, the delivery probability of n hops is given by

$$\varrho^{n}_{v,\tau}(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t_{\text{start}} \le t \le t_{\text{end}}} \left(1 - \varrho^{1}_{v,m}(t_{\text{start}}, t) \times \varrho^{n-1}_{m,\tau}(t, t_{\text{end}})\right). \tag{20}$$

Considering the routing in DTVN, a vessel has a packet p that needs to be delivered to $\zeta(p)$ with its remaining hops H and time-to-live TTL. When the vessel meets its neighbor v at time t_c, neighbor's delivery probability within $(H-1)$ hops needs to be calculated. The equation can be given by

$$\psi^{H-1}_{v,\varsigma(p)}(t_c, t_c + \text{TTL}) = 1 - \prod_{i=1}^{H-1} \left(1 - \varrho^{i}_{v,\varsigma(p)}(t_c, t_c + \text{TTL})\right). \tag{21}$$

5.3. Algorithm Description. The routing algorithm based on long-term trajectory prediction is described as follows. Assuming that each vessel can obtain recent historical knowledge and trajectory prediction model of other vessels, and vessels sharing the same channel resource set need to compete for limited useful links.

For packet list P in vessel v, the source vessel first calculates $\psi^{H-1}_{v,\varsigma(p)}(t_c, t_c + \text{TTL})$ of each p in P on v's neighbors.

The algorithm selects the relay node with the largest delivery probability. If links in the area are over link limitation, the algorithm will transfer the packet by probability rank to avoid channel collision.

When the packet p is copied and transferred, the system will update the copied packet's information. The algorithm would renew $H_{p\text{copy}}$. The packet will be abandoned if its TTL becomes 0. The packet p will stop forwarding when H is 0. The algorithm will repeatedly find the best relays for packets until packets in P are all delivered.

As an example, we show a small area of DTVN including vessel *a,b,c,d,e* in Figure 5. Vessel a met its neighbors b,c at time t_0 and wants to deliver its packet to e. The purple lines with arrow show the predicted trajectories of each vessel. Dotted line circles are vessels' expected location with confident degree, where $t_0 < t_1 < t_2$ and $t_2 - t_0 \ge \text{TTL}$. Therefore, the one-hop expected delivery probability between c and e can be calculated by $\varrho^{1}_{c,e} = E_{c,t2} \times E_{e,t2}$. Because c and e does not have other expected encounters. So $\psi^{H-1}_{c,e}(t_0, t_0 + \text{TTL})$ is $\varrho^{1}_{c,e}$. Vessels b and e have direct expected encounter at time t_2. So, the one-hop expected delivery probability between b and e can be calculated by $\varrho^{1}_{b,e} = E_{b,t2} \times E_{e,t2}$. Vessel b and e have a two-hop expected encounter situation where b meets d at t_1 and d meets e at $t2$. Hence, $\varrho_{2b,e} = (E_{b,t1} \times E_{e,t1})(E_{d,t2} \times E_{e,t2})$. According to equation (20), $\psi^{H-1}_{b,e} = 1 - (1 - \varrho^{1}_{b,e})(1 - \varrho^{2}_{b,e})$. In this case, α should compare its own expected delivery probability with $\psi^{H-1}_{b,e}$ and $\psi^{H-1}_{c,e}$ to decide next relay.

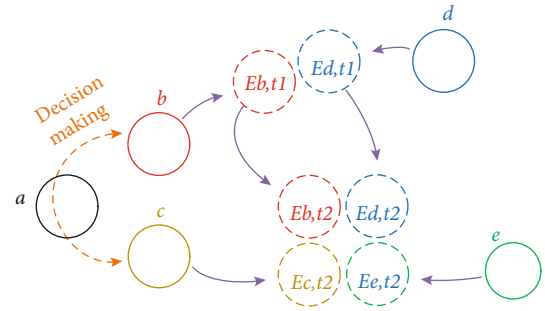


FIGURE 5: Illustration example of routing algorithm in DTVN.

6. Distributed Routing Algorithm

The previous section introduces the algorithm with global knowledge, including vessels' current location and mobility pattern. However, global knowledge is not available in distributed situations. Hence, we design the practical distributed algorithms with limited knowledge. In this section, we propose two distributed algorithms under fully distributed and cellular distributed situations.

6.1. TPR-Distributed Design. Compared with the global routing algorithm, the distributed algorithm further considers the actual application scenarios. So, the algorithm needs to make routing decisions utilizing incomplete knowledge of vessels and packets. In this situation, each vessel needs to have the ability to predict future contacts based on the received information and share its local information with meeting vessels. Each packet records its source node $\zeta(p)$, destination node ϱ_p, H, and TTL.

Therefore, each vessel can construct a local meeting graph in a period by exchanged information, and they can give prediction by sliding window and form the local link set $Ł_t$ in time t. When node_{cur} meeting node v on time t_c, the optimal target of packet p in one node is shown in Equation (8).

Information exchanging between vessels is vital in the fully distributed scenario. Thus, we design the data structure of exchanged information *inf* considering the efficiency and cost. It includes two parts: the current state sequence and prediction model, because other information, such as the evaluation table, contact graph, and future trajectory, could be calculated by these two parts.

As shown in Algorithm 1, the node n and its meeting node v both merge and update their local information. After

Input: $n, v, inf_n, inf_v, t_{cur}$.
Variables: n: the current node; v: the meeting node; inf_n, inf_v: information of node n, v; v_n: stored information of node n; t_{cur}: current time.
1: $v_n = v_n \cup v$.
2: Merge and Update: $inf_n, inf_v \leftarrow inf_n \cup inf_v$.
3: For each vessel in v_n, predict trajectory and evaluation table of all nodes between t_{cur} to $t_{cur} + TTL$ by their model and sliding window.
4: Construct contact graph based on inf_n.
5: For each $p \in P_n$, n calculate $\psi^{H-1}_{v,c(p)}(t_c, t_c + TTL)$.
6: Transmit each p to v when v has greater $\psi^{H-1}_{v,c(p)}(t_c, t_c + TTL)$.

Algorithm 1: TPR-distributed Algorithm.

that, both sides can construct a new local contact graph and form a local link set based on updated information. For each packet p in node n, p will be delivered to v, when v has greater $\psi^{H-1}_{v,c(p)}(t_c, t_c + \mathrm{TTL})$. If links in the area are over link limitation, the algorithm will transfer the packet by probability rank to avoid channel collision.

6.2. TPR-Cellular Design. Considering the cellular coverage range along the coast, we also design a cellular-based distributed algorithm called TPRCellular. Vessel within the cellular coverage area can provide communication with base stations in high efficiency. Hence, we assume the vessel in this area has the global knowledge of all the vessels. After moving out of the cellular coverage, each vessel's knowledge will stop updating. Nevertheless, they could make the routing decision based on predicted trajectory by prediction model and sliding window. Theoretically, by utilizing a sliding window, our model can endlessly predict future trajectory without considering the accuracy. So, this algorithm's routing performance will test our model's accuracy on long-term trajectory prediction. We assume that vessels do not exchange their knowledge when meeting with each other. Except for the predicted trajectory length, the TPR-Cellular algorithm is nearly the same as the global routing algorithm.

7. Performance Evaluation

We evaluate the performance of TPR-DTVN with metrics defined in Section 2.1. We first present the methodology for performance evaluation, introduce compared algorithms, and finally show simulation results.

7.1. Methodology and Experimental Setup. We perform trajectory-driven simulations to evaluate the performance of TPR-DTVN compared with Epidemic, Random Walk (RW), and Community-Based Routing (CBR).

We train the trajectory model and perform a simulation experiment on a Linux server, which is based on ubuntu 18.04 with 64 core CPUs and 4 pieces of 1080ti GPUs. We use a total of 5123 vessels that had appeared from May 2015 to May 2018. The first 24 months' data train trajectory prediction models in default. For each packet, we randomly pick its source and destination pairs which have at least one contact before, because vessels that have not encountered each other barely have communication requirements. The number of packets varies from 200 to 1800. 5 simulations are carried out over 5 separate fishing months (Oct 2017, Nov 2017, Dec 2017, Jan 2018, and Feb 2018) to achieve average results. The default system parameters are shown in Table 2.

Table 2: Default system parameters.

Parameter	Default value
Number of vessels	5123
Communication radius	20 km
Hmax	20
TTLmax	6 hours
Number of packets	200, 600, 1000, 1400, 1800

2.

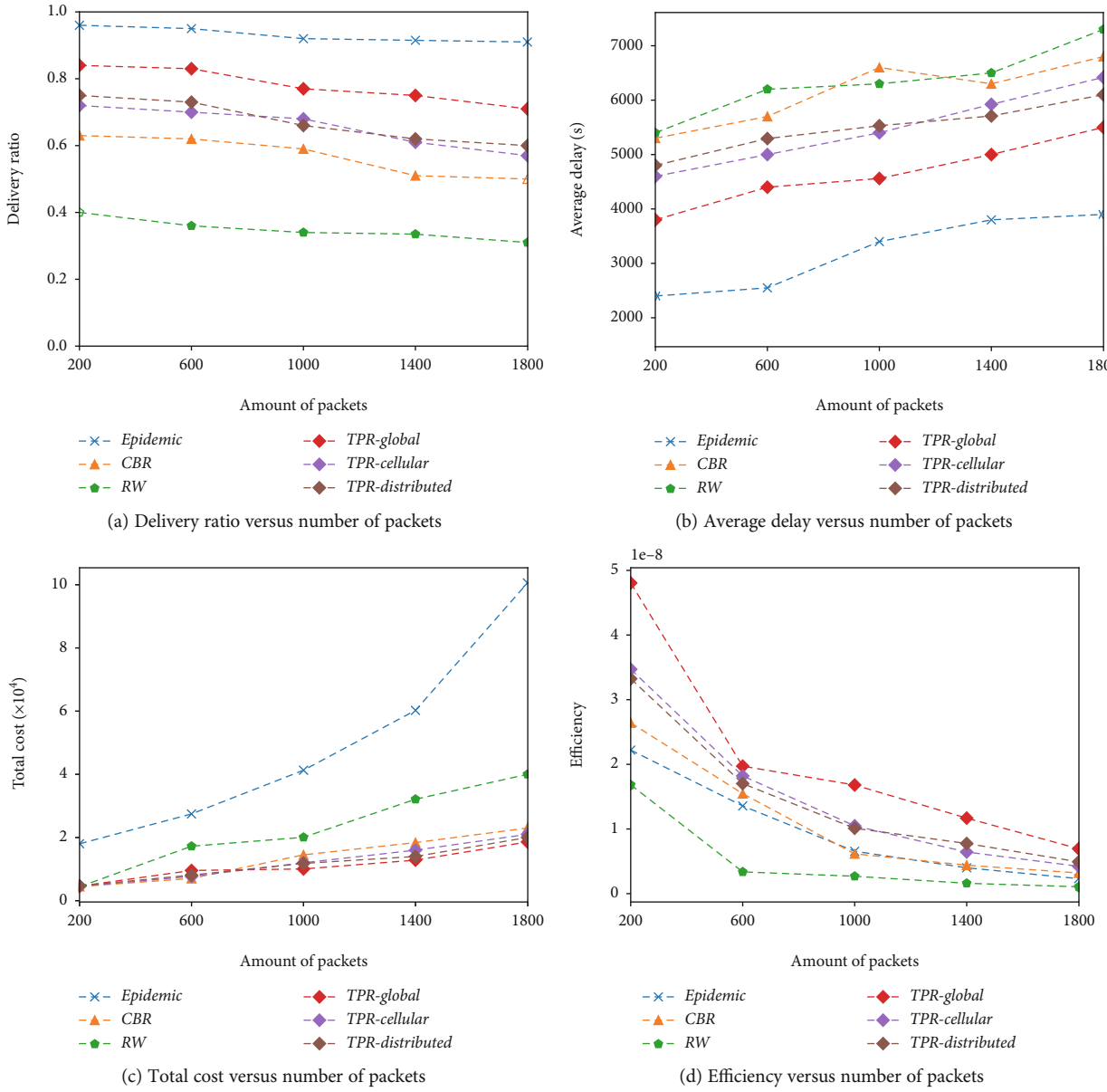
7.2. Compared Algorithms. Three compared algorithms are briefly introduced as follows:

(1) *Epidemic*. Packets are flooded throughout the network, which provides upper bound on delivery ratio and system cost with lower bound of delay

(2) *Random Walk (RW)*. This algorithm randomly decides whether to forward a packet and randomly selects a relay if there is more than one neighbor vessel. In simulation experiments, the probability of forwarding a packet is set to 40%. This algorithm represents the algorithms without knowledge

(3) *Community-Based Routing (CBR)* [12]. CBR utilizes vessels' historical contacts to form communities. Intercommunity betweenness centrality and familiarity are used to build a probability network for routing. This algorithm represents the algorithms with social knowledge

7.3. Comparative Results. We compare different routing algorithms' performance in delivery ratio, delay, cost, and efficiency. We vary the packet number from 200 to 1800 to simulate different network overhead. Figures 6(a) and 6(b) show six algorithms in terms of delivery ratio and average delay. The epidemic has the best performance in both metrics as expected. We can observe that TPR-Global, TPR-Distributed, TPRCellular, and CBR perform better than RW due to historical information. TPR-Global achieves a 21% higher



(a) Delivery ratio versus number of packets

(b) Average delay versus number of packets

(c) Total cost versus number of packets

(d) Efficiency versus number of packets

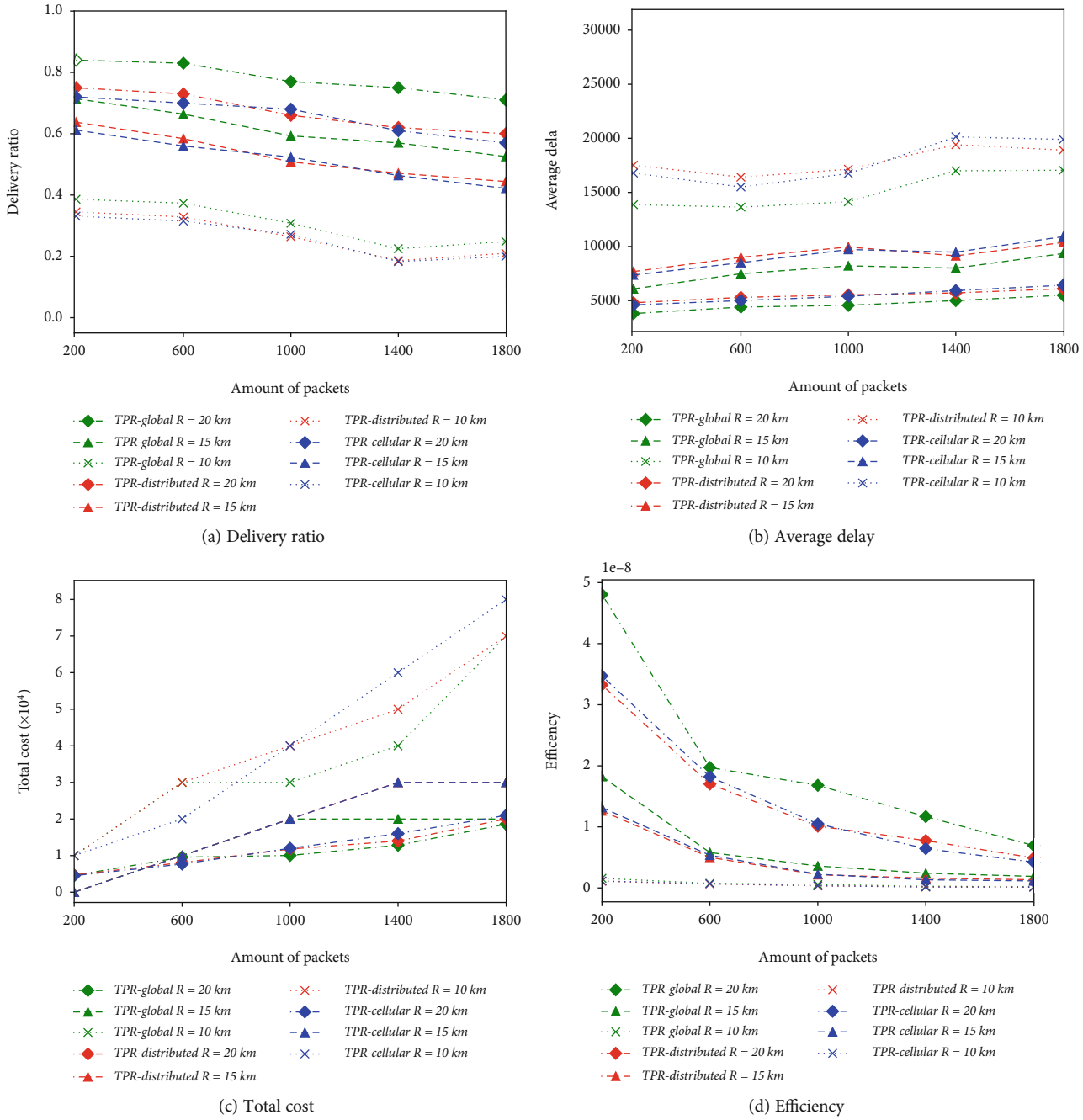
Figure 6: Comparative results of six algorithms.

delivery ratio than CBR with a nearly 70% delay of it. The reason is that CBR only uses contact information without considering the time factor. Although CBR can find who is familiar with the destination node, it cannot predict future contacts' approximate time. On the contrary, TPR-DTVN's probability calculation process is with the limitation of TTL , leading to more accurate results. Figure 6(c) shows the costs of six algorithms. TPR-DTVN and CBR have a lower cost than others, because wasted data transmission is avoided with historical knowledge guidance.

Figure 6(d) demonstrates the efficiency of all the algorithms. TPR-Global achieves the highest overall efficiency than the comparative algorithms. Because of the high cost of packet flooding, Epidemic's efficiency is uncompetitive. RW has the lowest efficiency due to its aimlessness. Also, TPRDistributed has a 16.25% higher efficiency than TPR-Cellular.

Although historical knowledge is different, TPRDistributed and TPR-Cellular have similar performance. Even if TPR-Cellular has the knowledge of all vessels, the vessels need to use longer prediction to get future trajectory based on its knowledge when the vessel moves out of cellular coverage. So the prediction accuracy is lower than that of TPR-Distributed. TPR-Distributed has limited knowledge, but its knowledge is up-to-date. These reasons make these two distributed algorithms have similar performance.

7.4. Impact of Communication Radius. We compare the performance of TPR-Global, TPR-Distributed, and TPR-Cellular with different communication radius, including 10 km,15 km, and 20 km. With the decrease of communication radius, we notice that the performance in all metrics decreases in Figure 7. Delivery ratio with $R = 10$ km has decrease 65% than that of $R = 20$ km. Meanwhile, the average



(a) Delivery ratio

(b) Average delay

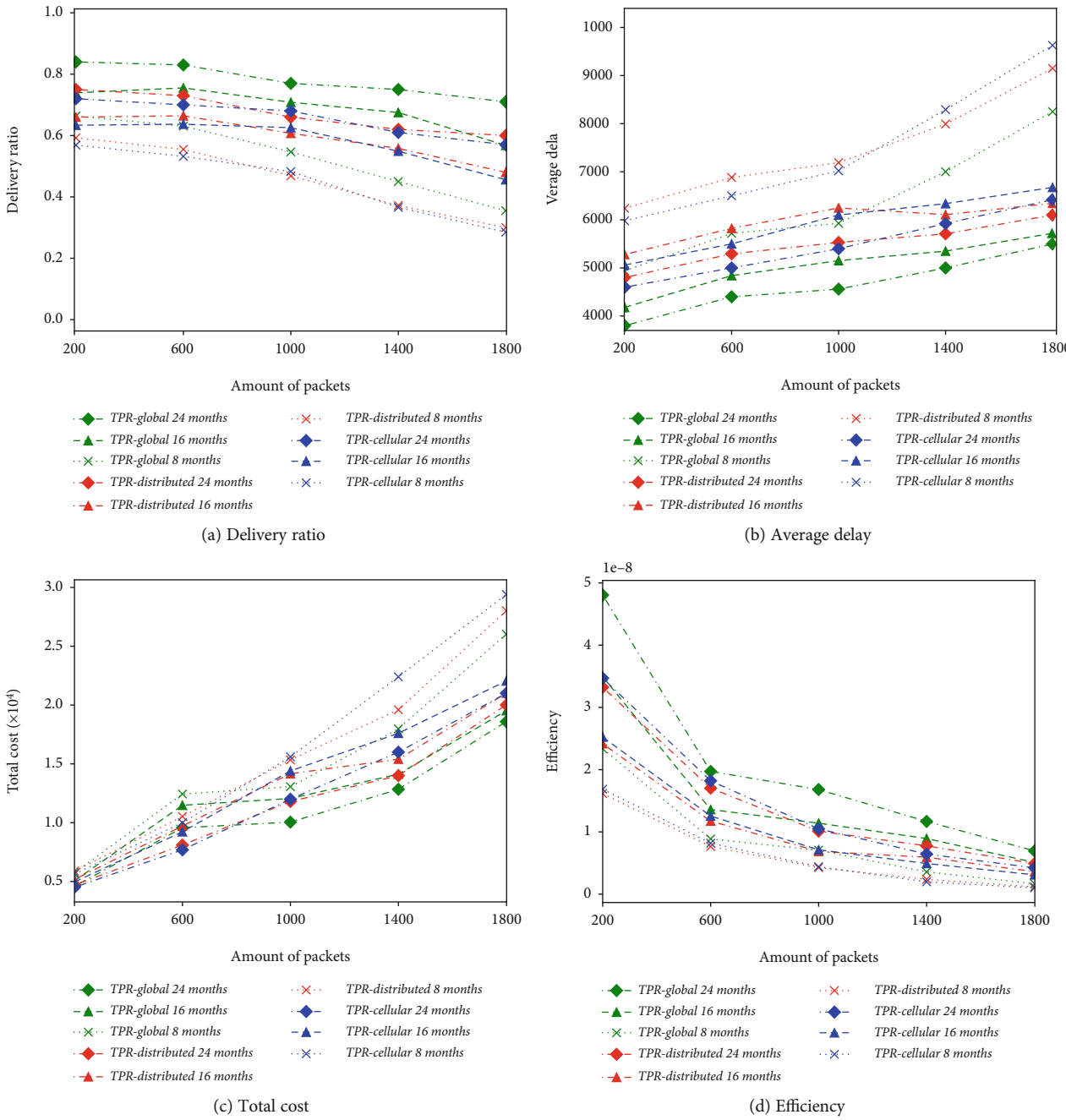
(c) Total cost

(d) Efficiency

Figure 7: Comparative results of TPR-DTVN with different communication radius.

delay is twice of $R = 20$ km. Compared with 20 km communication radius, these algorithms' efficiency in 15 km and 10 km average decrease 72.79% and 97.11%, respectively. The reason is that the communication opportunities of $R = 10$ km are not enough to achieve data delivery in this vessel density. WiMAX technology can achieve maritime communication with more than 40 km communication distance, which leads to fierce channel competition. Although the shorter communication radius will have less competition, it cannot grantee routing in our scenario. Considering the performance and channel resources, we set the communication radius to 20 km in our experiment.

7.5. Impact of Data Set Usage. The size of data set usage has a direct effect on TPR-Global and two distributed algorithms due to its dependency on the accuracy of the predicted trajectory. We compare the performance of different sizes of data sets, including the recent 24, 16, and 8 consecutive months. In Figure 8, with the decrease of data set, the performance of the three algorithms has notably declined. Compared train by 24 months data set, two algorithms' efficiencies in 16 months and 8 months average decrease 26.74% and 76.19%, respectively. We thought it is because the trajectory predicts model with 24 and 16 months data can learn complete mobility pattern of four seasons. 8 consecutive months cannot



(a) Delivery ratio

(b) Average delay

(c) Total cost

(d) Efficiency

Figure 8: Compare the performance of TPR-DTVN in different training datasets.

cover the mobility pattern of a season, which leads to poor performance. Therefore, we should choose more than 12 consecutive months of data to train our model to obtain excellent algorithm performance.

8. Relate Works

DTVN, which utilizes WiMAX to achieve vessel-to-vessel or vessel-to-coast communication, is complementary to maritime wireless mesh networks [6, 23]. There have been several researches applying traditional routing algorithms in MANETS. Mohsin et al. evaluate traditional algorithms' delivery ratio under different node densities and mobility behaviors to validate the feasibility of DTVN [8]. Lambrinos et al. use real small-scale traces to simulate different protocols' performance and identify where they fall short from the perspective of our particular domain [24].

Some studies apply fixed mobility models or encounter models for the design of DTVN routing algorithms. These algorithms are often suitable for vessels with stable mobility behavior, such as container ships, cruise ships, and waste-dumping vessels. Raj et al. presume that all vessels obey the Gaussian-Markov mobility model [7]. The forwarding metrics are determined for routing on the basis of this assumption. Lanepost [25] and Mar-DTN [26] are standard graphic-based optimal routing algorithms. These algorithms

build an opportunistic routing graph based on each vessel's fixed expected routes and use the graph for forwarding decisions. Liu et al. [27] and Qin et al. [28] study delay-tolerant routing problems in the seaway model that all vessels are moving in fixed seaways. Vessels predict the speed and arrival time based on proper models and utilize them for decisions.

Mining personalized characteristics of each node in the network are more beneficial for routing algorithm performance than fixed model [13, 29, 30]. Some vessels' moving patterns, like fishery vessels, are easier to be affected by objective factors, which cannot directly apply fixed models. Nodes in the network use neighbors' personalized models for routing decisions. These models need to be dynamically updated based on historical data. FBR and CBR [12] dynamically construct communities based on vessels' recent contacts. Based on unique social relationships, vessels build a probability network for efficient packet delivery.

9. Conclusion

This paper proposed a set of routing algorithms called TPRDTVN for efficient data transmission in vessel networks. TPR-DTVN uses a forwarding metric that characterizes the expected delivery probability of relay nodes. To address the challenge of long-term trajectory prediction, we design a Bi-LSTM-based trajectory prediction model and an evaluation method to get the practicable predicted trajectory. Then, we evaluate expected delivery probability with predicted information. Extensive trace-driven simulations show that our algorithms can achieve a higher delivery ratio with lower cost and transmission delay.

Data Availability

The data is not available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

C. Liu and Y. Li contributed equally to this work and should be regarded as co-first authors. R. Jiang and Y. Du are corresponding authors.

Acknowledgments

This research was supported by the National Natural Science Foundation of China Grant No. 61902367, 62072287, and 41976185; the China Postdoctoral Science Foundation 2020M682240; the Natural Science Foundation of Shandong Grant No. ZR2018BF006; and the National Key R&D Program 2016YFC1401900.

References

[1] K. Metcalfe, N. Breheret, E. Chauvet et al., "Using satellite ais to improve our understanding of shipping and fill gaps in ocean observation data to support marine spatial planning," *Journal of Applied Ecology*, vol. 55, no. 4, pp. 1834–1845, 2018.

[2] M. C. Domingo, "An overview of the internet of underwater things," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879–1890, 2012.

[3] P. Carnie, S. Kenny, E. Browell, F. Cheng, H.-C. Fang, and A. Incecik, "Global marine trends 2030: implications for naval ship technology," in *Pacific 2013 International Maritime Conference: The commercial maritime and naval defence showcase for the Asia Pacific*, p. 241, Australia, 2013.

[4] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G. B. Huang, "Exploiting ais data for intelligent maritime navigation: a comprehensive survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–24, 2016.

[5] F. Li, D. Yu, H. Yang, J. Yu, and X. Cheng, "Multi-armed-banditbased spectrum scheduling algorithms in wireless networks: a survey," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 24–30, 2020.

[6] C. Liu, Y. Li, R. Jiang, F. Hong, and Z. Guo, "Oceannet: a low-cost large-scale maritime communication architecture based on d2d communication technology," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, New York, NY, USA, 2019.

[7] D. Raj, M. V. Ramesh, and S. Duttagupta, "Delay tolerant routing protocol for heterogeneous marine vehicular mobile adhoc network," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 461–466, Kona, HI, USA, 2017.

[8] R. J. Mohsin, J. Woods, and M. Q. Shawkat, "Density and mobility impact on manet routing protocols in a maritime environment," in *2015 Science and Information Conference (SAI)*, pp. 1046–1051, London, UK, 2015.

[9] J. Dhivvya, S. N. Rao, and S. Simi, "Towards maximizing throughput and coverage of a novel heterogeneous maritime communication network," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 1-2, New York, NY, USA, 2017.

[10] S. Qin, G. Feng, W. Qin, Y. Ge, and J. S. Pathmasuntharam, "Performance modeling of data transmission in maritime delay-tolerantnetworks," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1109–1114, Shanghai, China, 2013.

[11] T. Yang, H. Liang, N. Cheng, R. Deng, and X. Shen, "Efficient scheduling for video transmissions in maritime wireless communication networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4215–4229, 2014.

[12] Q. Bing, R. Jiang, and F. Hong, "Exploiting social network characteristics for efficient routing in ocean vessel ad hoc networks," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, London, United Kingdom, 2019.

[13] Y. Zhu, Y. Wu, and B. Li, "Trajectory improves data delivery in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 1089–1100, 2013.

[14] C. Liu, Z. Sun, J. Liu, H. Huang, Z. Guo, and Y. Feng, "Vstp: vessel spatio-temporal contact pattern detection based on mapreduce," *Eurasip Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017.

[15] C. Liu, Y. Liu, Z. Guo, X. Wang, and S. Guo, "Mr-similarity: parallel algorithm of vessel mobility pattern detection," in *Big*

Data Computing and Communications, pp. 225–235, Springer, 2016.

[16] C. Liu, S. Guo, Y. Feng, F. Hong, H. Huang, and Z. Guo, "L-vtp: long-term vessel trajectory prediction based on multi-source data analysis," *Sensors*, vol. 19, no. 20, article 4365, 2019.

[17] P. Borkowski, "The ship movement trajectory prediction algorithm using navigational data fusion," *Sensors*, vol. 17, no. 6, article 1432, 2017.

[18] S. Guo, C. Liu, Z. Guo, Y. Feng, F. Hong, and H. Huang, "Trajectory prediction for ocean vessels base on k-order multivariate markov chain," in *Wireless Algorithms, Systems, and Applications*, pp. 140–150, Springer, 2018.

[19] Z. Sun, K. Hu, T. Hu, J. Liu, and K. Zhu, "Fast multi-label low-rank linearized svm classification algorithm based on approximate extreme points," *IEEE Access*, vol. 6, pp. 42319–42326, 2018.

[20] Y. Zhou and L. Shao, "Vehicle re-identification by adversarial bi-directional lstm network," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 653–662, Lake Tahoe, NV, USA, 2018.

[21] F. Altche and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 353–359, Yokohama, Japan, 2017.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] M.-T. Zhou, V. D. Hoang, H. Harada et al., "Triton: highspeed maritime wireless mesh network," *IEEE Wireless Communications*, vol. 20, no. 5, pp. 134–142, 2013.

[24] L. Lambrinos, C. Djouvas, and C. Chrysostomou, "Applying delay tolerant networking routing algorithms in maritime communications," in *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(-WoWMoM)*, pp. 1–6, Madrid, Spain, 2013.

[25] X. Geng, Y. Wang, H. Feng, and L. Zhang, "Lanepost: lane-based optimal routing protocol for delay-tolerant maritime networks," *China Communications*, vol. 14, no. 2, pp. 65–78, 2017.

[26] P. Kolios and L. Lambrinos, "Optimising file delivery in a maritime environment through inter-vessel connectivity predictions," in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 777–783, Barcelona, Spain, 2012.

[27] C. Liu, Z. Guo, F. Hong, and K. Wu, "Dcep: data collection strategy with the estimated paths in ocean delay tolerant network," *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, Article ID 518439, 2014.

[28] S. Qin, G. Feng, W. Qin, Y. Ge, and J. S. Pathmasuntharam, "Performance evaluation of data transmission in maritime delaytolerant-networks," *IEICE Transactions on Communications*, vol. 96, no. 6, pp. 1435–1443, 2013.

[29] Y. Yuan, F. Li, D. Yu et al., "Fast fault tolerant sampling via random walk in dynamic networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems*, pp. 536–544, Dallas, TX, USA, 2020.

[30] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, and X. S. Shen, "Zoom: scaling the mobility for fast opportunistic forwarding in vehicular networks," in *2013 Proceedings IEEE INFOCOM*, pp. 2832–2840, Turin, Italy, 2013.