

Research Article

Optimization of LoRa SF Allocation Based on Deep Reinforcement Learning

Han Zhong ^{1,2}, Lei Ning ¹, Junsong Wang ¹, Siliang Suo ³, and Liming Chen ³

¹College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China

²College of Applied Technology, Shenzhen University, Shenzhen, China

³Electric Power Research Institute Co., Ltd, Guangzhou, China

Correspondence should be addressed to Lei Ning; ninglei@sztu.edu.cn

Received 9 July 2022; Revised 4 September 2022; Accepted 23 September 2022; Published 12 October 2022

Academic Editor: Mingqian Liu

Copyright © 2022 Han Zhong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

LoRa is an IoT communication technology that realizes ultra-long-distance transmission through spread spectrum modulation. However, its ultra-long-distance transmission also sacrifices the corresponding rate, and data conflicts are prone to occur when the number of nodes is large. In this article, we investigate various types of data collisions in LoRa wireless work, most of which are affected by Spreading Factor (SF) assignment. At present, the distribution of the SF for LoRa in the industry is mostly based on Min-airtime and Min-distance. In the case of a large number of nodes, the data collision between nodes will increase sharply. This paper proposes a SF redistribution scheme under limited network resources, in order to improve the terminal capacity of the LoRa gateway. First, the problem of minimizing the data collision rate without expanding gateway or network resources is presented. Specifically, the reallocation of SF with increasing number of terminals is studied. Finally, considering the randomness of the data sent by the terminal, SF redistribution schemes based on deep reinforcement learning (DRL) are developed. The simulation results show that the collision rate of the proposed SF redistribution scheme is nearly 30% lower than Min-airtime and Min-distance, and its total energy consumption is close to Min-distance. Therefore, the proposed SF redistribution scheme can effectively improve the gateway capacity of LoRa wireless network.

1. Introduction

In recent years, the Internet of Things industry has developed rapidly, existing mobile cellular communication technology cannot meet the communication requirements of long-distance, low power consumption, and large connection of IoT node equipment [1, 2]. In this context, low power wide area network (LPWAN) [3] came into being, which is a general term for a communication technology suitable for long distance, low power consumption, low bandwidth, and multiconnection IoT connections [4]. LPWAN includes LoRa, NB-IoT [4], RPMA [5, 6], Sigfox [7, 8], LTE-M [9, 10], and other wireless communication technologies [11, 12]. Among them, it can be divided into two categories according to whether authorization is required. In the unlicensed frequency band, LoRa has been widely used in the field of IoT since its invented due to its long transmission

distance and low power consumption. Compared with NB-IoT, which requires operator authorization, its on-demand deployment and low deployment cost also make it favored by many organizations and companies with ad hoc [13] network needs. Especially in scenarios with weak signal, long transmission distance and low power consumption requirements, LoRa has more advantages than other communication technologies [14].

LoRa [15, 16] achieves super anti-interference and long-distance transmission through spread spectrum modulation technology. This technology trades bandwidth for sensitivity [17] and is used in communication technologies such as WiFi [18, 19] and Zigbee [20]. LoRa modulation is characterized by maximizing sensitivity, even approaching the limit of Shannon's theorem [21]. While LoRa achieves such long-distance transmission, it also sacrifices some speed. When the anti-interference ability

is stronger and the transmission distance is longer, the amount of data that can be transmitted per unit time is less. In this process, the main parameter of the SF plays an important role [22].

At present, the mainstream method for setting the SF is based on Min-distance and Min-airtime [23]. The two methods determine the optimal SF selection based on the optimal selection of distance and transmission time, respectively. Therefore, neither of the two methods considers the correlation between nodes when the node data is large or the data sending time is close, so the data conflict will increase as the number of nodes increases. How to reduce the data collision rate between LoRa nodes has become a major concern in this paper.

In this paper, we develop a DRL-based LoRa SF allocation optimization method to dynamically optimize the SF allocation of LoRa nodes, thereby reducing the occurrence of collisions between nodes. To showcase the efficiency, we compare the proposed DRL-based method with traditional node feature-based SF assignment methods. Contributions can be summarized as follows:

- (i) We first study the performance characteristics of nodes under different SFs for the LoRa collision problem. For different SFs, virtual simulation scenarios are built to simulate the characteristics of transmission under different SFs. Through comparative analysis, it is shown that the optimal rate does not necessarily have the lowest collision rate in the selection of the SF
- (ii) Aiming at the existing algorithm for selecting SF based on feature, this paper proposes to consider the influence of channel environment on SF. An optimization algorithm based on DRL framework is proposed. The algorithm considers selecting the optimal SF by combining the node's own characteristics and channel environment information. Considering that the gateway cannot know whether there is a collision between nodes, we consider whether the node is retransmitted to determine whether the node collides or loses packets. Finally, we redesign the state and action parameters of the algorithm
- (iii) On the simulation platform, we compare the performance of the DRL-based SF optimization algorithm with the feature-based Min-distance and Min-airtime algorithms. When the number of nodes reaches 1000, the optimization algorithm we propose collides with each other. The rate is reduced by nearly 30%, and the total energy consumption is close to the Min-distance

The rest of the paper is organized as follows. Section 2 presents related works. Section 3 provides the problem formulation and system model. Section 4 proposes DQN based SF allocation. Section 5 elaborates the numerical results, and finally, Section 6 summarizes the conclusion and future work.

2. Related Work

At the beginning of the invention of LoRa, the node uses the pure ALOHA [24] protocol to send data, and the node does not perform channel detection but sends it directly. In this way, as the number of terminals increases or the number of sent packets increases, the probability of packets from multiple terminals colliding on the channel is greatly increased. Since the LoRa mechanism is too simple, on the basis of LoRa, the LoRa Alliance has launched the LoRaWAN protocol [25, 26]. In LoRaWAN, a duty cycle is proposed to constrain the node to occupy the channel all the time, thus avoiding data conflict to a certain extent, but in the case of a large amount of data, due to the duty cycle, the data delay will be increased. In the case of a large amount of data concurrency, its advantages are not obvious. In addition, LoRaWAN also introduces a CAD [27] mechanism to reduce the probability of LoRa conflict, but this undoubtedly increases the power consumption of LoRa. The most important thing is that this does not improve the situation of LoRa nodes crowding the channel and increase the capacity of the gateway. At present, the Class A mode in LoRaWAN still uses pure ALOHA. According to calculations, the channel utilization of the pure ALOHA protocol is only 18.4%, and most nodes send collisions in the channel. Therefore, how to improve LoRa to increase its gateway capacity and reduce the collision rate has become a major research point in the industry.

To this end, the industry has also done a lot of work and research on LoRa collision optimization. Edward et al. [28] expect to increase the capacity of the gateway by introducing Interleaved Chirp Spreading LoRa- (ICS-LoRa-) based modulation. In [29], The authors optimize the transmission parameters of a LoRaWAN system in high density smart city traffic environment using golden section search and parabolic interpolation. Floris et al. [30] used ns-3 to simulate and analyze the LoRa network. The analysis shows that increasing gateway density can ameliorate but not eliminate this effect, as stringent duty cycle requirements for gateways continue to limit downstream opportunities. Reynders et al. [31] present a scheme to efficiently optimize the packet error rate fairness inside a LoRaWAN cell. This is achieved by optimizing the power and SF for each node while avoiding near-far problems by allocating distant users to different channels. In [32], Abdelfadeel et al. present results of a study of the data rate fairness among nodes within a LoRaWAN cell. In order to make the rate of each node more relatively fair, they secondly develop a transmission power control algorithm to balance the received signal powers from all nodes regardless of their distances from the gateway for a fair data extraction. But this algorithm only considers the case where the set of nodes is close to the network.

As a new intelligent decision-making algorithm, AI has been widely used in many fields. In network resource scheduling, many scholars have also introduced AI for scheduling and decision optimization. Jiang et al. [33] used reinforcement learning to optimize the throughput and transmission time interval of NB-IoT. Yang et al. [34] utilize deep neural network (DNN) to configure optimized NOMA for network

resource management. The DNN not only greatly improves the computational efficiency but also improves the summation rate of the system. In [35], a DQN method is used to control the Handover (HO) procedure of the User Equipments (UEs) by well capturing the characteristics of wireless signals interference and network load. Experimental results show that the proposed scheme can reduce HO rate and guarantee the system throughput, which is better than the traditional HO scheme.

In this article, we consider using the reinforcement learning algorithm to further improve and optimize the distance-based optimization algorithm and redistribute the SF of the nodes to reduce the collision rate of the channel with the growth of nodes or data explosion. Finally, increase the capacity of channel nodes under a single gateway. In Section 3, we will first introduce LoRa's communication collision model and the problem formulation will be presented as follows.

3. System Model and Problem Analysis

3.1. LoRa Communication Model. On the basis of the LoRa physical layer, the LoRa Alliance released the LoRaWAN protocol to adapt to the LoRa physical layer. In LoRaWAN [36], Class A mode must be implemented by default, and it mainly uses the ALOHA protocol for data transmission [37]. The basic idea is that each node can send data frames at any time, and then monitor the channel to see if there is a conflict. If a conflict occurs, then the node will wait for a random period of time to retransmit until the retransmission is successful. The collision process is shown in Figure 1.

In LoRa, there are four main parameters that generally affect the conflict between data, namely center frequency, SF, Bandwidth(BW), and Coding Rate(CR). Through reasonable configuration, there can be a variety of mutually orthogonal combinations to avoid conflicts [38]. The four parameters are described as follows

- (1) *CR.* The coding rate is the ratio of the useful part of the data stream. LoRa uses cyclic error correction coding for forward error detection and error correction. However, using this method will generate transmission overhead. The specific overhead is shown in Table 1
- (2) *Center Frequency.* The frequency in the middle of the filter passband, in LoRa, the corresponding frequency needs to be set according to laws and regulations. There is a random frequency hopping mechanism in LoRaWAN. This mechanism is mainly to deal with the regulations of LoRa duty cycle, so as to solve the problem of transmitting large data packets. This mechanism can also be effective in reducing collisions at certain times
- (3) *BW.* The signal bandwidth is to limit the lower frequency and upper frequency of the signal allowed to pass through the channel. In LoRa, increasing

the BW can improve the transmission rate of the payload, but it will also reduce the receiver sensitivity while reducing the transmission time

- (4) *SF.* LoRa spread spectrum modulation is realized by representing each bit of data in the payload information with multibit chip information. Since different SFs are orthogonal to each other, the SFs must be known in advance in a group of transceiver links

The modulation modes of LoRa signals are orthogonal to each other under different SF and BW combinations, and data transmission can be performed using Code Division Multiple Access (CDMA). In the same channel, if the BW is unchanged, the multichannel orthogonal data can be transmitted without interfering with each other by changing the SF. The SF ranges from 5 to 12, and a total of 8 addresses are used. In LoRaWAN, a total of six addresses are used from 7 to 12, and the rate corresponding to each multiple access is also different. The speed will affect its air flight time, and the air flight time can be calculated by the following formula.

$$ToA = N_{sym} * \frac{2^{SF}}{BW}. \quad (1)$$

ToA is the air time, and N_{sym} is the number of symbols. Due to the different modulation parameters, the calculation of the number of symbols is also different, and the specific calculation is given in the following formula.

For $SF = 5$, and $SF = 6$,

$$N_{sym} = N_{sympream} + 6.25 + 8 + \text{ceil}\left(\frac{\max(N_{bit}, 0)}{4 * SF}\right), \quad (2)$$

$$N_{bit} = 8 * N_{bp} + N_{bitCRC} - 4 * SF + N_{symheader}.$$

For other SF,

$$N_{sym} = N_{sympream} + 4.25 + 8 + \text{ceil}\left(\frac{\max(N_{bit}, 0)}{4 * SF}\right), \quad (3)$$

$$N_{bit} = 8 * N_{bp} + N_{bitCRC} - 4 * SF + 8 + N_{symhead},$$

When CRC is turned on, $N_{bitCRC} = 16$, otherwise it is 0. $N_{symhead} = 20$ in explicit header mode, 0 in recessive header mode. N_{bp} represents the byte payload. The air interface time required for different SFs and different packet lengths can be calculated through the above calculation formula. The air time is shown in Figure 2.

If the air time of transmission is longer, it means that the time of occupying the channel is longer. At this time, if other nodes in LoRa transmit with the same SF, the data of the two nodes will interfere with each other, which will cause the gateway to fail to receive. At this time, the node will retransmit when no response is received. The larger the SF, the smaller the capacity of the channel and the greater the probability of collision. How to make a reasonable distribution of the SF becomes a problem we are concerned about. In the

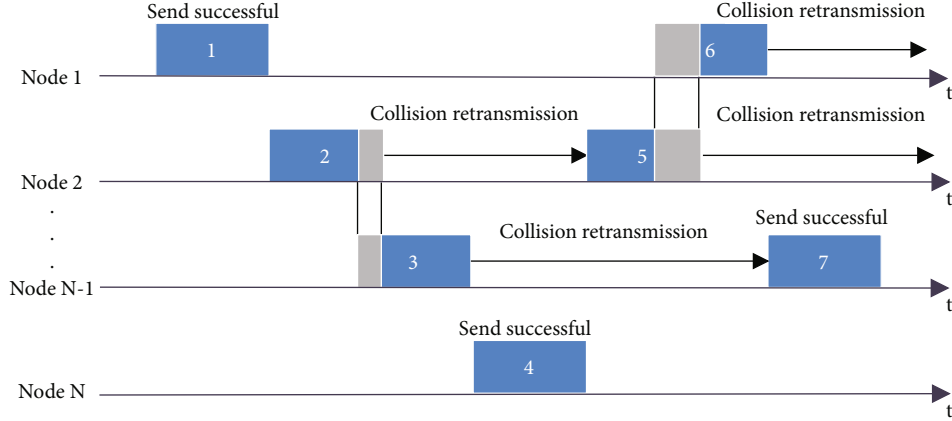


FIGURE 1: The collision process of the pure ALOHA protocol.

TABLE 1: CR Parameters.

CR	Cyclic coding rate	Overhead ratio
1	4/5	1.25
2	4/6	1.5
3	4/7	1.75
4	4/8	2

next subsection, we will further analysis of the success rate of data reception by LoRa's SF.

3.2. Problem Analysis and Description. In LoRa, there are four main factors that affect the collision of node data, namely, Frequency Collision, SF Collision, Power Collision, and Timing Collision. SF is an important parameter of LoRa communication transmission. The effect of setting different SFs on collision is particularly obvious. In order to verify the impact of LoRa nodes using different SFs on the success rate of network data transmission, this paper uses the Python simulation simulator LoRaSim to simulate the collision of LoRa nodes and verify the impact of different SFs on the success rate of LoRa gateway data reception. Nodes are randomly distributed within a radius of 2 kilometers with the gateway as the center. The spreading factor SF $\in [7, 8, 9, 10, 11, 12]$, which ensures that each node can communicate with the LoRa gateway normally. The packet sending interval of each node is 5 minutes; the packet load length is 20 bytes; the bandwidth is 125 kHz, and the number of gateway channels is 1. The total simulation time is 2 hours. Finally, under different SFs, as the number of nodes increases, the graph of the successful data reception rate is obtained, as shown in Figure 3.

In Figure 3, there are differences in the data transmission success rates corresponding to different SFs under the same number of nodes, which shows that the low SF has a higher data transmission success rate, while the high SF has a lower data transmission success rate. As the number of nodes increases, the success rate of data transmission is also affected by the SF. When SF = 12, the number of nodes exceeds 300; the success rate of data transmission is already

lower than 30%, which seriously affects the reliability of the network.

Since a higher SF has a stronger anti-interference ability, the transmission of the same size of data needs to occupy the channel for a longer time, and the energy consumption is also higher. Additional energy consumption is also required for data retransmission due to packet loss or data collision. Figure 4 shows the energy consumption of data transmission with different SFs. Therefore, when the data is reachable, a lower SF is generally preferred for transmission.

The optimal SF transmission has its advantages, but in the case of a large number of nodes, if many nodes select the optimal SF, the collision rate of the optimal SF will increase. Due to the orthogonal relationship between the SFs, the LoRa gateway can receive a variety of node data with different SFs or bandwidths at the same time. The communication multiple access that LoRa can be expressed is as follows:

$$N_{\text{channel}} = N_{\text{SF}} * \sum N_{\text{BW}} \quad (4)$$

$$\text{BW} \in [62.5, 125, 250, 500]$$

In the actual deployment situation, letting all nodes select the optimal SF may increase the collision rate instead. With the help of LoRa's feature that different SFs can be demodulated at the same time, how to allocate SFs more reasonably, so as to maximize the utilization of network resources and reduce the collision rate of nodes can be described as follows:

$$\text{Node}_i^{\text{SF}} = \text{argmin} \sum \left(\beta * N_i^{\text{collision}} + \gamma * N_i^{\text{lost}} \right) \quad (5)$$

The formula describes that the optimal SF allocated should be the smallest collision rate and no packet loss. In Formula (5), $i \in M$, M is the number of nodes; β and γ are reduction coefficients. In order to prevent packet loss as much as possible, gamma weights should be larger.

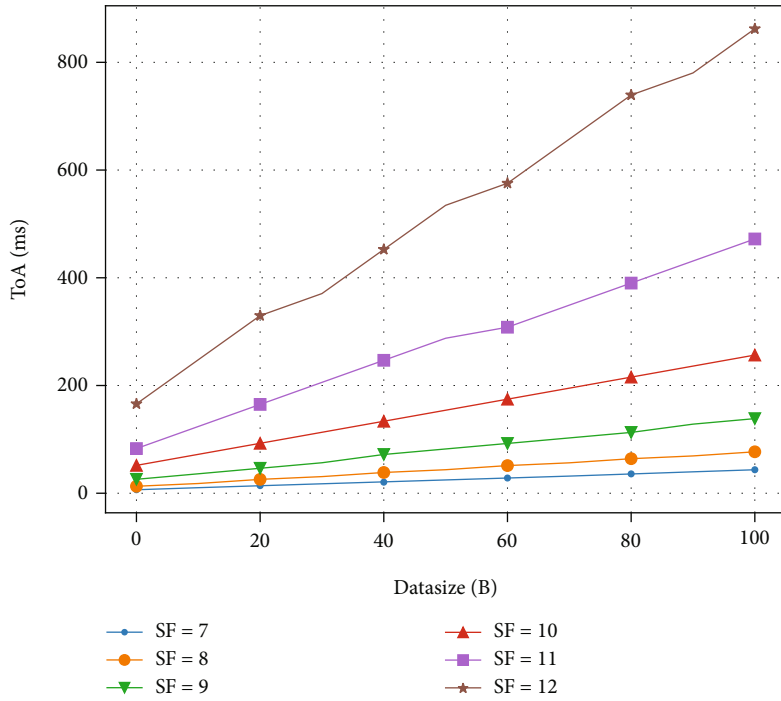


FIGURE 2: Time required to send data with different SF.

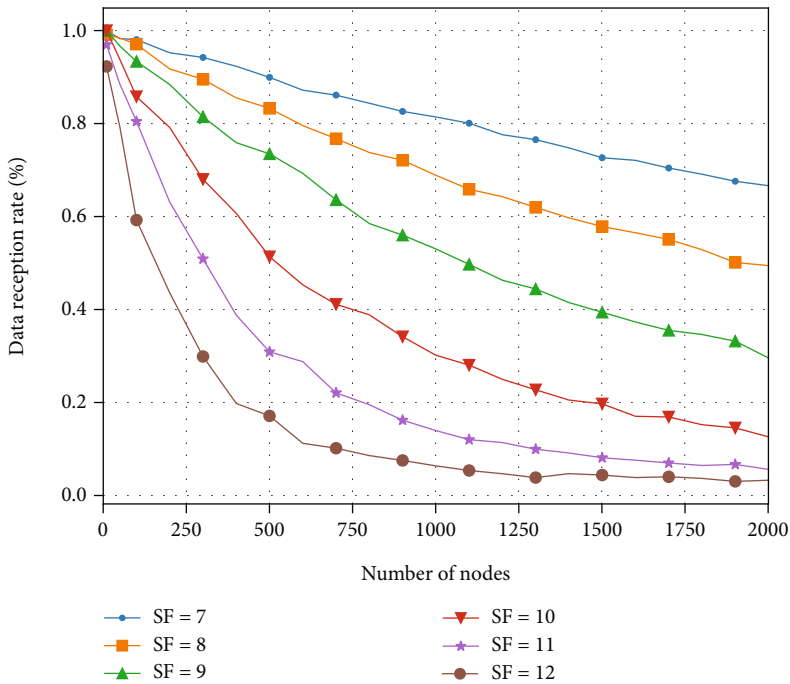


FIGURE 3: Successful reception rates for different SF.

4. SF Allocation Based on DRL

A reinforcement learning algorithm is an AI algorithm that optimizes itself according to changes in the environment. It is mainly composed of agents and environments. The agent observes some state parameters required by itself from the environment, and gives corresponding action parameters according to the state parameters. At present, reinforcement

learning algorithms have shined in many fields, such as game AI, autonomous driving, etc.

In order to optimize the distribution of the SF, in this section, the reinforcement learning algorithm combined with the distance optimization algorithm of LoRa is proposed to be embedded in the LoRa network server. Figure 5 shows a schematic diagram of the Deep Reinforcement Learning (DRL) algorithm embedded in the network server.

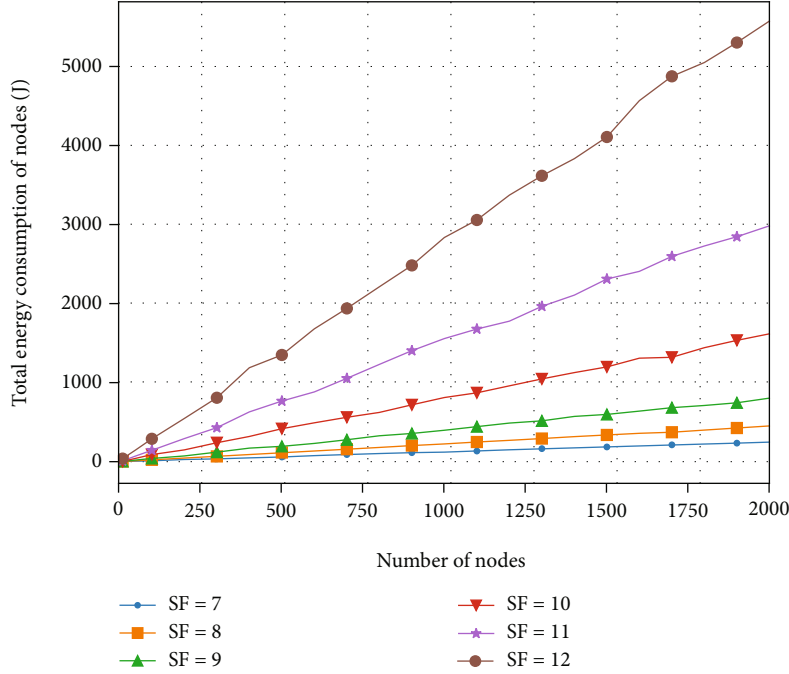


FIGURE 4: Energy consumption under different SF.

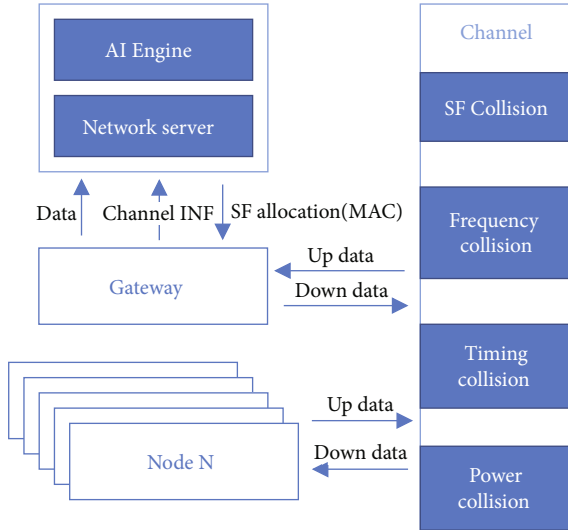


FIGURE 5: AI embed in the network server.

In Figure 5, the AI algorithm is designed to be embedded in the network server, and the LoRa gateway performs the forwarding function of the node. The LoRa node sends uplink data based on the ALOHA mechanism. Only when the data does not collide in the channel can the LoRa gateway successfully receive the data sent by the node (Data) and forward it to the network server, which also includes the channel environment information (Channel INF); after the network server receives the data, it will parse it; through the environment information recorded by each node (whether retransmission occurs, sending time, etc.), it determines if there is a collision between nodes. The AI algorithm will give the corresponding adjustment strategy according to the node information and channel information. The strategy

will be forwarded through the gateway, and the gateway will send the MAC command when the node's receiving window is opened, so as to adjust the channel parameter settings of each node to maximize the use of each channel resources and reduce the collision rate of data.

4.1. Deep Reinforcement Learning Algorithm Model. The deep reinforcement learning algorithm plans to use the Deep Q Network (DQN) algorithm [39]. The DQN algorithm is a method of approximating the value function of Q-learning through a neural network [40]. Q-learning is a model-free reinforcement learning technique proposed by Watkins in 1989. For a given environmental state, it can have relatively good operational expectations without the need for an environmental model. At the same time, it can handle random transitions and reward issues without adjustment. It has been shown that, for any finite MDP, Q-learning will eventually find an optimal policy, i.e., starting from the current state, the expected value of the total return over all successive steps is the maximum achievable [41]. Before learning begins, Q is initialized to a possibly arbitrary fixed value. Then, at each time t , the agent chooses an action a_t , gets a reward R_t and enters a new state S_{t+1} , and the Q value is updated. Its core is the value function iteration process, namely,

$$\begin{aligned} Q(s_t, a_t) &\leftarrow Q(s_t, a_t) + \alpha \cdot U_{t+1} U_{t+1} \\ &= r_t + \gamma \max_{\pi} Q(s_{t+1}, a_t) - Q(s_t, a_t). \end{aligned} \quad (6)$$

However, when encountering a large number of state spaces or a continuous state, Q-learning will face the disaster of dimensionality or the difficulty of storing rough retrieval, so a neural network is introduced to approximate the value

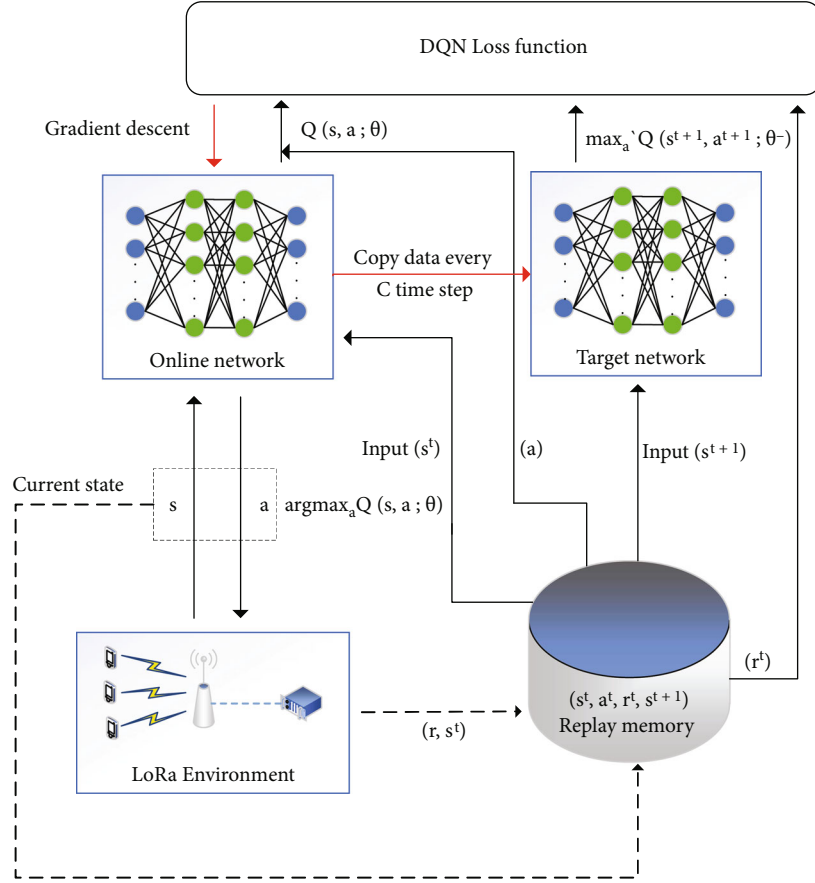


FIGURE 6: DQN algorithm framework.

function [42, 43]. After the introduction of the neural network, the whole process becomes how to determine θ to approximate the value function. In this paper, gradient descent is used to minimize the loss function to debug the network weight θ . At this time, the Loss Function is defined as follows:

$$L_i(\theta_i) = E_{(s,a,r,s') \sim U(D)} \left[\left(r + \gamma \max_a Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right]. \quad (7)$$

In the given formula, we use two neural networks; one is called the Q neural network, and the other is called the target neural network. The purpose of introducing the target neural network is to reduce the correlation between the current Q value and the target Q value, thereby improving the stability of the algorithm. Specifically, θ_i^- is the target network parameter of the i iteration, and θ_i is the current Q neural network parameter.

In addition, experience playback is also introduced to store the past state, and the method of random sampling from the experience pool is adopted to update the neural network parameters, thereby breaking the correlation between data and improving the utilization of data. The learning process of DQN is shown in Figure 6.

4.2. DQN Parameters Design. In this section, we will introduce how DQN integrates and interacts with LoRa's network server, including some specific parameters of the environment state as well as the action parameters generated by DQN.

4.2.1. State. The LoRa network server receives the packet data in the LoRaWAN format from the gateway. The data packet includes the node ID, packet length, transmission interval, SF, encoding rate, and bandwidth. Each node has a unique ID number. From the node number information, the network server can know the location of each node and the distance dis from the forwarding gateway. At the same time, it is also possible to know whether retransmission has occurred from the information of the data packet. If the data packet is retransmitted, it means that the data packet has a data collision or is lost. Then, the state of the environment can be represented as a 7-dimensional vector.

$$s_i^{node} = \{s_{id}, s_{dis}, s_{sf}, s_{bw}, s_{len}, s_{interval}, s_{col}\}. \quad (8)$$

s_{dis} represents the distance between the gateway and the node device. This data is added to the network server when the node is initialized to join the network for the first time. The SF is an integer of 7-12, and the BW optional frequency bands are 125, 250, and 500.

```

Initialize LoRa node based on LoRaWAN
Initialize replay memory D to capacity N
Initialize action-value function Q with random weights  $\theta$ 
Initialize target action-value function  $Q^-$  with weights  $\theta^- = \theta$ 
For episode=1,M do do
  Initialize and choose state
   $S_{sf,bw}$  from LoRa network server(MAC)
  For  $t = 1, T$  do
    With probability  $\varepsilon$  select a random action  $a_t$  otherwise select  $a_t = \operatorname{argmax}_a Q(s_t, a; \theta)$  execute action  $a_t$  in emulator
    Observe reward  $r_t$  and next state  $s_{t+1}$ 
    Store experience  $(s_t, a_t, r_t, s_{t+1}) \rightarrow D$ 
     $D \rightarrow (s_i, a_i, r_i, s_{i+1})_{(i \in B)}$ 
    If episode terminates at step  $j+1$  then
       $y_j = r_j$ 
    Else
       $y_j = r_j + \gamma_a' Q^-(s_{j+1}, a'; \theta^-)$ 
    End if
    Perform a gradient descent step on  $(y_j - Q(s_j, a_j; \theta))^2$  with respect to the network parameters  $\theta$ 
    If batch size  $\geq$  memory capacity then
      Update  $Q^- \leftarrow Q$ 
    End if
  End for
End for

```

ALGORITHM 1: Procedure of deep Q network.

4.2.2. *Action.* The action parameter consists of two parts, namely, the selection of the SF and the selection of the bandwidth. There are a total of 18 combinations of the two to choose from. So the output of the neural network is an 18-dimensional vector, and the one with the largest value is selected from the 18-dimensional vector.

$$a_t = \operatorname{argmax}\{a_{sf1,bw1}, a_{sf1,bw2}, a_{sf1,bw3}, \dots, a_{sf12,bw3}\}. \quad (9)$$

4.2.3. *Reward.* In reinforcement learning, the agent's goal is formally represented as a special signal, called a reward, which is passed to the agent through the environment. At each instant, the reward is a single scalar value. In this paper, the expected maximum goals are the collision rate between nodes and the packet loss rate of nodes. The reward function in this paper is set as follows:

$$r_i^t = -\frac{N_{col} + N_{lost}}{1 + N_{rec}} - \frac{f_{lost} + f_{col}}{(1 + N_{rec})/(1 + N_{col} + N_{lost})}, \quad (10)$$

where

$$f_{lost/col} = \begin{cases} 1, & \text{either collision or packet lost,} \\ 0, & \text{neither collision nor packet lost.} \end{cases} \quad (11)$$

The goal of RL is to maximize the cumulative discounted reward functions by finding an optimal policy. We then define long-term reward V_i as the accumulated and dis-

TABLE 2: Environmental parameters.

Parameter	Value
Working channel (MHz)	471.5
Max BS receive	8
γ	2.08
d_0	40
L_{pl}	127.41
Bandwidth(kHz)	125, 250, 500
SF	7, 8, 9, 10, 11, 12
C/R	4/5

TABLE 3: DQN algorithm parameters.

Parameter	Value
Greedy policy ε	0.9
Batch size	256
Reward discount γ	0.9
Target update frequency	100
Memory capacity	5000
Actions	18
States	7

counted reward.

$$V_i = \sum_{t=0}^{T-1} \gamma^t r_i^t. \quad (12)$$

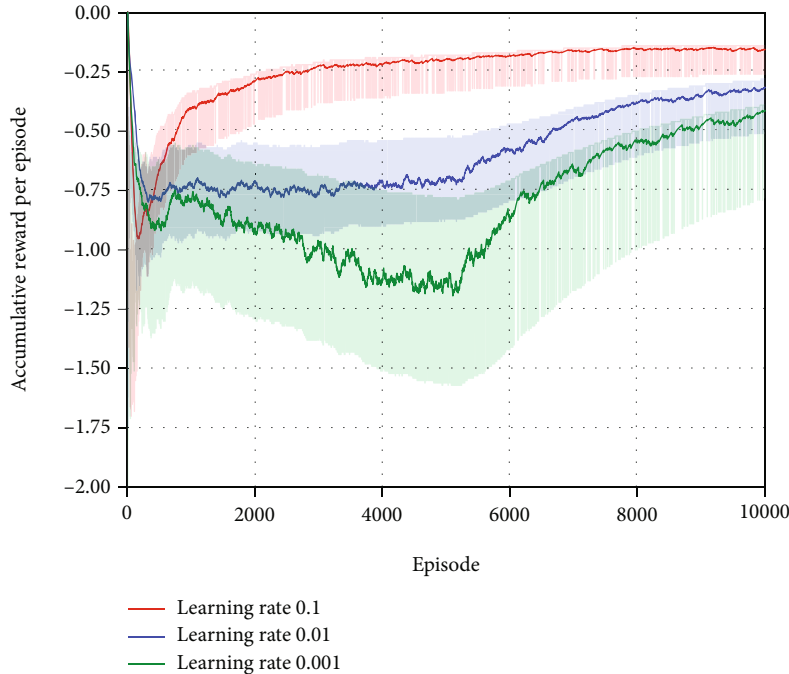


FIGURE 7: Different learning rate of training DQN.

γ is the reduction coefficient, which belongs to $[0, 1]$. When the reduction coefficient is 0, it means that only current interests are considered. When the reduction coefficient is larger, it means that longer-term interests are considered.

At the end of this section, we present the procedure of the DQN-based SF allocation optimization algorithm. The algorithm will be shown in Algorithm 1.

5. Performance Evaluation

In this section, we will first introduce the LoRa network environment and the settings of related parameters, and secondly, we will optimize the algorithm model for training based on the LoRaSim simulator. Finally, we will give a performance comparison of LoRa based on different algorithms.

5.1. Parameter Settings. We will use the Python-based LoRa simulator LoRaSim to simulate the LoRa network communication environment. In LoRaSim, the process of communication between multiple nodes and gateways is simulated through the SimPy-based discrete event library, and each node and gateway is maintained by a thread. When each thread is simulating sending packets, a collision function will be used to simulate whether a collision occurs. The settings of the collision function include SF collision, frequency collision, and time collision. In order to be as close to the real communication environment as possible, the parameters of the network environment will be set according to the LoRaWAN protocol. The specific environment parameter settings are shown in Table 2.

For the neural network model, considering that the number of states and actions are not large, the two neural network models in DQN adopt a three-layer network model

with 50 neurons in each layer. Other algorithm parameters are shown in Table 3.

In the training simulation environment of the algorithm model, we set 1500 nodes and initialize the default allocation of SF = 7; the data volume of each node is 50 bytes; the sending interval is 5 minutes, and the total simulation time is 2 hours. In order to prevent overfitting, the number of rounds is taken as the first 10000. Three different learning rates are set, and the training curve of the algorithm is presented in Figure 7.

As shown in Figure 7, when the learning rate is 0.001 and 0.01, the training requires a large number of rounds to achieve a good effect, and after the algorithm convergence, its collision rate is still higher than that of the algorithm with a learning rate of 0.1. Therefore, we choose a learning rate of 0.1.

5.2. Algorithm Performance and Comparison. In the LoRa simulation experiment in this article, we will select two widely used algorithms for comparison, namely, Min-distance and Min-airtime. The Min-distance allocation strategy allocates the SF according to the range of the RSSI value received by the gateway. A low RSSI value corresponds to a high SF, and a high RSSI value corresponds to a low SF. This method uses a low SF to improve the success rate of data transmission but does not fully consider the orthogonality between SFs. It assigns almost all nodes to SF = 7 and SF = 8. The Min-airtime allocation strategy is to adaptively select the minimum combination of air transmission time corresponding to bandwidth, SF and coding rate according to the situation of the node itself. Compared with Min-distance, this deployment has better flexibility.

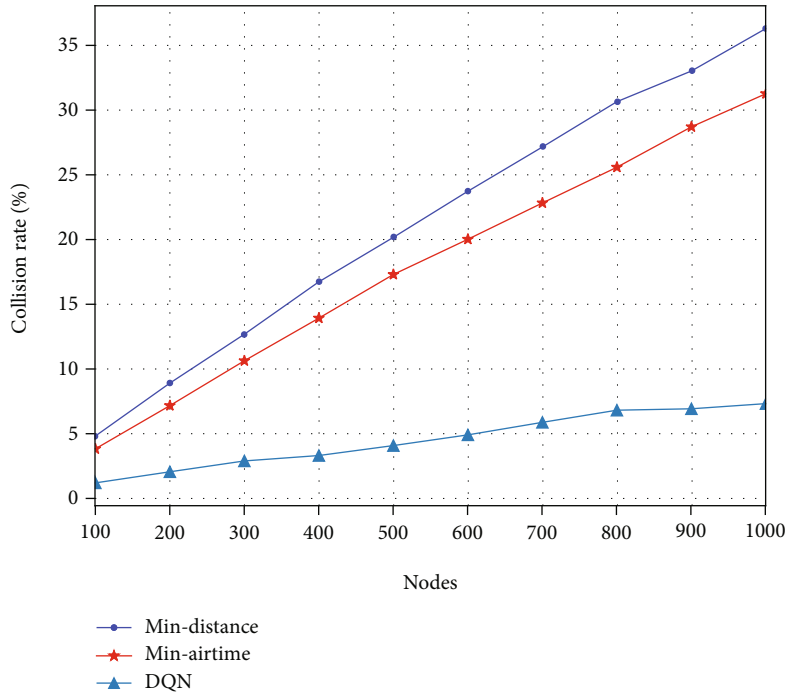


FIGURE 8: LoRa collision rate comparison.

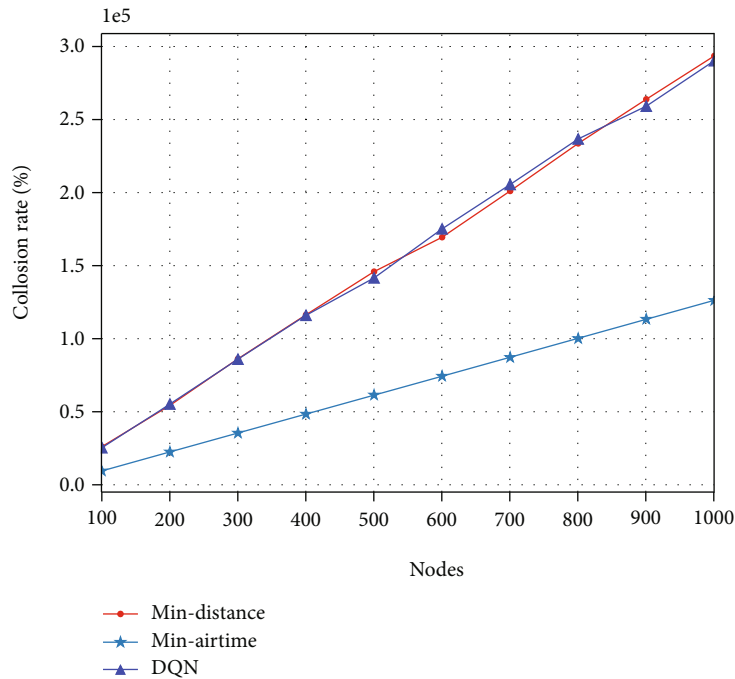


FIGURE 9: Total energy consumption.

In this experiment, the number of nodes in the range from 100 to 1000 will be selected, and the value will be taken every 100 nodes. The node location information is initialized and fixed when entering the network. The transmission time interval is set to 5 minutes, and the packet length is set to 50 bytes. The total simulation time is 12 hours. The DQN algorithm proposed in this paper and the comparison of the two algorithms are shown in Figure 8.

In Figure 8, the DQN algorithm redistributes the SF, and as the number of nodes increases, the data collision rate is significantly reduced compared with the other two algorithms. At 1000 nodes, its collision rate is reduced by 24%. During the same period, node collisions tend to be clustered. If a large number of collisions occur in the short-term SF = 7, due to the need for retransmission, the collision of data cannot be alleviated in the short-term. And DQN considers

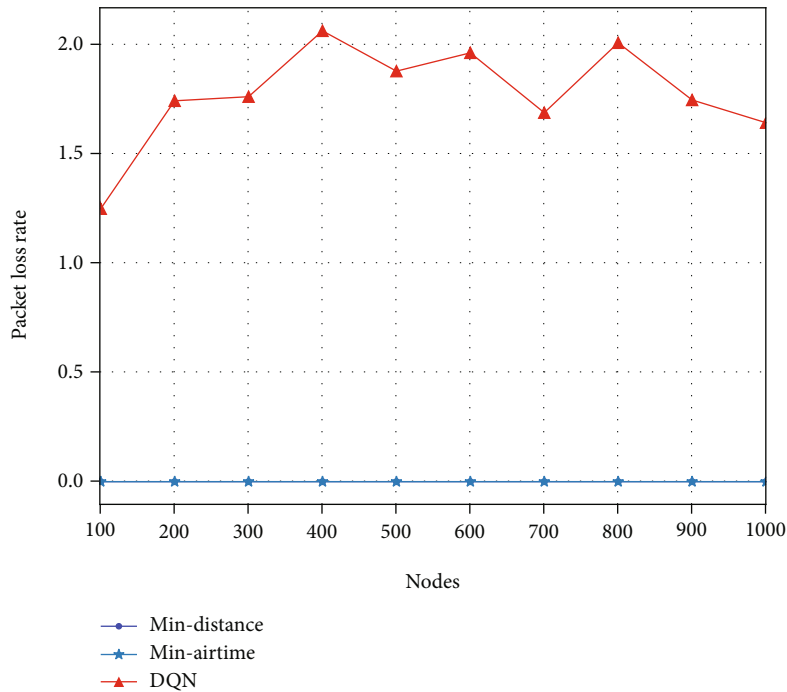


FIGURE 10: LoRa packet loss rate comparison.

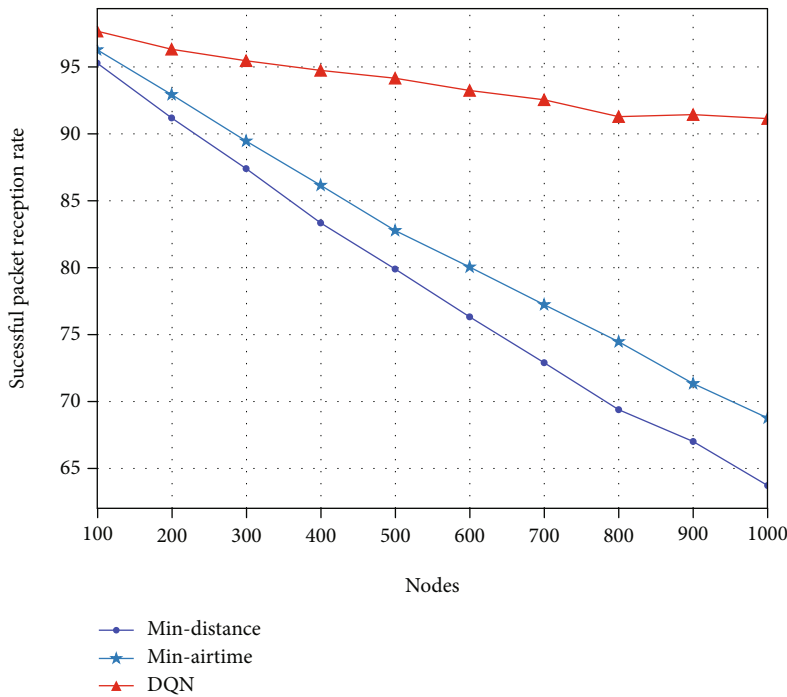


FIGURE 11: Successful data reception rate.

using more spreading factors, including some SF options with poor rate, in order to reduce the collision explosion for a period of time. Therefore, compared with the Min-distance and Min-airtime methods, when there are more nodes, DQN can play a more significant role in reducing collisions.

In Figure 9, the energy consumption comparison chart under the three methods is shown. Due to the assignment of a higher SF, a larger energy consumption is required to complete the data transmission. Therefore, its energy consumption is slightly higher than that of Min-airtime, but it is basically the same as that of Min-distance. SF is

proportional to energy consumption. Although DQN reduces the collision of nodes, from the perspective of energy consumption, energy consumption increases due to the allocation of higher SF.

DQN also brings a certain degree of packet loss after adjusting the SF. The packet loss problem is due to the assignment of unreasonable SFs to nodes that suffer from strong interference. Min-airtime and Min-distance allocate the SF by calculating in advance to ensure that the node will not lose packets when sending data, and then select the optimal SF. It can also be found in the simulation experiment that the packet loss rate of the two methods is 0. DQN will search for better SF when assigning SF. If an unreasonable SF is selected, packet loss will occur. In the DQN algorithm, the packet loss rate of a node has nothing to do with the number of nodes but is related to the DQN learning exploration rate. The packet loss phenomenon generally occurs in the stage of DQN learning the environment. As the algorithm observes and learns from the environment, the total packet loss rate is not more than 3%. The packet loss rate comparison chart is shown in Figure 10.

In Figure 11, we give the success rate of the gateway receiving data under different methods. The main factor that affects the success rate of Min-airtime and Min-distance data receiving is the conflict between data. In DQN, the algorithm uses the SF. The adjustment has resulted in a certain degree of packet loss but greatly reduced the conflict between data. With the increasing number of nodes, DQN data reception success rate is better than Min-airtime and Min-distance. In the case of 1000 nodes, the data reception success rate of the DQN algorithm is 27% higher than that of Min-airtime and 28% higher than that of Min-distance.

6. Conclusion

Aiming at the problem that LoRa is prone to data collision in large-scale node scenarios, this paper proposes an optimization algorithm for LoRa SF allocation based on deep reinforcement learning. By analyzing the performance of different LoRa SF and the impact on the collision rate, it is found that the collision between nodes can be effectively reduced by reasonably configuring the SF of LoRa nodes. We developed a LoRa SF allocation optimization algorithm based on the DQN algorithm. According to the analysis of the LoRa SF, the environmental state information that has a major impact on the collision rate was selected, and the action information was combined and configured. The simulation results show that as the number of nodes increases, the algorithm we developed can effectively reduce the data collision rate between LoRa nodes, and the energy consumption is close to Min-distance.

Due to exploration characteristics of the DRL algorithm, a small part of the packet loss phenomenon will occur in LoRa nodes. In the future, we will expect to further optimize the LoRa packet loss situation. And with the maturity of satellite-based cooperative communication, the optimization for joining multimode communication will also become a research direction. This kind of multiobjective optimization is very challenging and should be solved in the future.

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 sponsored by SZTU-Winoble Cooperation Research Project (No. 2021010802015 and No. 20213108010030); Scientific Research Capacity Improvement Project from Guangdong Province (No.2021ZDJS109); SZTU Experimental Equipment Development Foundation (No. JSZZ202102007).

References

- [1] A. S. Shaker, "A survey of smart buildings and homes using low-powerwide-area network (LoRa WAN)," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1–7, Istanbul, Turkey, 2020.
- [2] L. Da Xu, W. He, and S. Li, "Internet of things in industries: a survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [3] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: an overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
- [4] S. Popli, R. K. Jha, and S. Jain, "A survey on energy efficient narrowband internet of things (NB-IoT): architecture, application and challenges," *IEEE Access*, vol. 7, pp. 16739–16776, 2019.
- [5] M. I. Nashiruddin, M. T. B. Sihotang, and M. A. Nugraha, "Random phase multiple access network planning for smart city deployment in urban area," in *2021 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, pp. 232–237, Bandung, Indonesia, 2021.
- [6] M. I. Nashiruddin, S. Winalisa, and M. A. Nugraha, "Random phase multiple access network for public internet of things in Batam island," in *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 311–316, Semarang, Indonesia, 2021.
- [7] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of cellular LPWAN technologies for IoT deployment: Sigfox, LoRaWAN, and NB-IoT," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 197–202, Athens, Greece, March 2018.
- [8] A. Lavric, A. I. Petrariu, and V. Popa, "Long range SigFox communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, pp. 35816–35825, 2019.
- [9] "LTE-advanced in 3GPP Rel -13/14: an evolution toward 5G | IEEE Journals & Magazine | IEEE Xplore," <http://iee.org/abstract/document/7432169>.

- [10] M. Mezzavilla, M. Zhang, M. Polese et al., "End-to-end simulation of 5G mmWave networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2237–2263, 2018.
- [11] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J.-C. Prévotet, "Internet of mobile things: overview of LoRaWAN, DASH7, and NB-IoT in LPWANs standards and supported mobility," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1561–1581, 2019.
- [12] S.-H. Hwang and S.-Z. Liu, "Survey on 3GPP low power wide area technologies and its application," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, pp. 1–5, Singapore, 2019.
- [13] V. Geetha, S. Aithal, and K. Chandra Sekaran, "Effect of mobility over performance of the ad hoc networks," in *2006 International Symposium on Ad Hoc and Ubiquitous Computing*, pp. 138–141, Mangalore, India, 2006.
- [14] O. Georgiou and U. Raza, "Low power wide area network analysis: can LoRa scale?," *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.
- [15] A. J. Wixted, P. Kinnaird, H. Larijani et al., "Evaluation of LoRa and LoRaWAN for wireless sensor networks," in *2016 IEEE SENSORS*, pp. 1–3, Orlando, FL, USA, 2016.
- [16] P. Edward, S. Elzeiny, M. Ashour, and T. Elshabrawy, "On the coexistence of LoRa- and interleaved chirp spreading LoRa-based modulations," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–6, Barcelona, Spain, 2019.
- [17] P. Gkotsiopoulos, D. Zorbas, and C. Douligeris, "Performance determinants in LoRa networks: a literature review," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1721–1758, 2021.
- [18] D. Suh, H. Ko, and S. Pack, "Efficiency analysis of WiFi off-loading techniques," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3813–3817, 2016.
- [19] "WiFi and bluetooth security | IEEE Courses | IEEE Xplore," <https://ieeexplore.ieee.org/courses/details/EDP457>.
- [20] G. Pan, J. He, W. Qin, R. Fang, J. Cao, and D. Liao, "Automatic stabilization of Zigbee network," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 224–227, Chengdu, China, 2018.
- [21] G. Zhu, C.-H. Liao, T. Sakdejayont, I. W. Lai, Y. Narusue, and H. Morikawa, "Improving the capacity of a mesh LoRa network by spreading-factor-based network clustering," *IEEE Access*, vol. 7, pp. 21584–21596, 2019.
- [22] J. P. S. Sundaram, D. Wan, and Z. Zhao, "A survey on LoRa networking: research problems, current solutions, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 371–388, 2020.
- [23] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating inter-network interference in lora networks," in *In EWSN '17 Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, pp. 323–328, ACM Press, New York, 2017.
- [24] F. Baccelli, B. Blaszczyszyn, and P. Muhlethaler, "An Aloha protocol for multihop mobile wireless networks," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 421–436, 2006.
- [25] A. Lavric, A. I. Petrariu, and L. Anchidin, "Internet of things software defined radio technology for LoRaWAN wireless communication: a survey," in *2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pp. 1–4, Bucharest, Romania, March 2021.
- [26] N. P. Manchev, K. K. Angelov, P. G. Kogias, and S. M. Sadinov, "Development of multichannel LoRaWAN gateway for educational applications in low-power wireless communications," in *2019 IEEE XXVIII International Scientific Conference Electronics (ET)*, pp. 1–4, Sozopol, Bulgaria, September 2019.
- [27] F. Lehmann, "Joint user activity detection, channel estimation, and decoding for multiuser/multiantenna OFDM systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8263–8275, 2018.
- [28] P. Edward, M. El-Aasser, M. Ashour, and T. Elshabrawy, "Interleaved chirp spreading LoRa as a parallel network to enhance LoRa capacity," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3864–3874, 2021.
- [29] A. L. Emmanuel, X. Fernando, F. Hussain, and W. Farjow, "Optimization of spreading factor distribution in high density LoRa networks," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–5, Antwerp, Belgium, 2020.
- [30] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, 2017.
- [31] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, 2017.
- [32] Q. Khaled, V. C. Abdelfadeel, and D. Pesch, "Fair adaptive data rate allocation and power control in LoRaWAN," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 14–15, Chania, Greece, June 2018.
- [33] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.
- [34] N. Yang, H. Zhang, K. Long, H. Hsieh, and J. Liu, "Deep neural network for resource management in NOMA networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 876–886, 2020.
- [35] M. Wu, W. Huang, K. Sun, and H. Zhang, "A DQNbased handover management for SDN-enabled ultra-dense networks," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, Victoria, BC, Canada, 2020.
- [36] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of Lo-RaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.
- [37] L. Beltramelli, A. Mahmood, P. Österberg, and M. Gidlund, "LoRa beyond ALOHA: an investigation of alternative random access protocols," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3544–3554, 2021.
- [38] M. Bor and U. Roedig, "LoRa transmission parameter selection," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 27–34, Ottawa, ON, Canada, June 2017.
- [39] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, p. 1054, 1998.
- [40] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

- [41] Q. Lan, Y. Pan, A. Fyshe, and M. White, "Maxmin Q-learning: controlling the estimation bias of Q-learning," 2020, <https://arxiv.org/abs/2002.06487>.
- [42] F. Skomorokhov and G. Ovchinnikov, "Reinforcement learning for assignment problem," 2020, <http://arxiv.org/abs/2011.03909>.
- [43] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: a brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.